

**Artificial Intelligence
in Intelligent Learning Environments
and other Applications**

Research in Computing Science

Series Editorial Board

Editors-in-Chief:

Grigori Sidorov (Mexico)
Gerhard Ritter (USA)
Jean Serra (France)
Ulises Cortés (Spain)

Associate Editors:

Jesús Angulo (France)
Jihad El-Sana (Israel)
Alexander Gelbukh (Mexico)
Ioannis Kakadiaris (USA)
Petros Maragos (Greece)
Julian Padget (UK)
Mateo Valero (Spain)

Editorial Coordination:

María Fernanda Ríos Zacarias

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 111**, septiembre 2016. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. : 04-2005-121611550100-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor responsable: *Grigori Sidorov, RFC SIGR651028L69*

Research in Computing Science is published by the Center for Computing Research of IPN. **Volume 111**, September 2016. Printing 500. The authors are responsible for the contents of their articles. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research. Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

Artificial Intelligence in Intelligent Learning Environments and other Applications

Carlos Alberto Reyes García (ed.)



Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"



Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2016

ISSN: 1870-4069

Copyright © Instituto Politécnico Nacional 2016

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional “Adolfo López Mateos”, Zacatenco
07738, México D.F., México

<http://www.rcs.cic.ipn.mx>

<http://www.ipn.mx>

<http://www.cic.ipn.mx>

The editors and the publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

All rights reserved. No part of this publication may be reproduced, stored on a retrieval system or transmitted, in any form or by any means, including electronic, mechanical, photocopying, recording, or otherwise, without prior permission of the Instituto Politécnico Nacional, except for personal or classroom use provided that copies bear the full citation notice provided on the first page of each paper.

Indexed in LATINDEX, DBLP and Periodica

Printing: 500

Printed in Mexico

Editorial

El presente volumen de la revista *Research in Computing Science* titulado “Inteligencia Artificial en Ambientes de Aprendizaje Inteligentes y otras Aplicaciones” ofrece una amplia variedad de trabajos tanto terminados como en desarrollo de temas relacionados con tutoriales inteligentes y otros ambiente donde métodos de la inteligencia computacional combinados con métodos multimediales son usados como herramientas de soporte al proceso educativo. En general, el objetivo de todos los artículos incluidos va dirigido a reforzar la educación no presencial y a distancia por medio de internet o con apoyo de computadoras. Es así que en este libro se incluyen temas como el que describe una herramienta para el apoyo automático de la calificación de exámenes, un trabajo que describe el proceso de diseño de recurso educativo abierto apoyado por un sistema experto orientado a la Enseñanza-Aprendizaje de Base de Datos. Se presenta también una aplicación de procesos Markovianos para recomendar acciones pedagógicas óptimas en tutores inteligentes. Hacia un modelo de interfaces multimodales adaptables a los canales de aprendizaje en aplicaciones colaborativas como apoyo a la educación. Siguiendo con el tema se encuentran trabajos como la presentación de una base de conocimientos para asistir el aprendizaje de la programación y un sistema tutor afectivo como apoyo al aprendizaje de las matemáticas y en este caso utilizando técnicas de Gamificación (introducción de juegos). Este trabajo se complementa con la plataforma de reconocimiento multimodal de emociones que posteriormente se plantean utilizar en un sistema tutor afectivo para la enseñanza de lógica algorítmica y programación. Los desarrollos de ambientes inteligentes de aprendizaje continúan presentándose con un método para evaluar afecto en un entorno de aprendizaje para Java. Los temas afines orientados a aplicaciones masivas continúan con la presentación de un sistema de reconocimiento multimodal de emociones relacionadas al aprendizaje en dispositivos móviles. Siguiendo con aplicaciones de juegos para apoyar el proceso educativo se plantean métodos para hacer una revisión sistemática de técnicas de Gamificación en aplicaciones educativas inteligentes, y un tema afín relacionado con una metodología para la creación de objetos de aprendizaje adaptables al estilo de aprendizaje. Y aunque no directamente relacionado con el proceso educativo, pero si en apoyo al mejoramiento de la educación se encontrara un modelo difuso para evaluar la pertinencia de un programa educativo de nivel superior. Complementando los temas de ambientes inteligentes de aprendizaje se incluyen trabajos que aplican metodologías inteligentes a otros campos como el que propone un método para reducir el espacio de búsqueda en entornos virtuales. Y el dedicado a la identificación automática de marcadores argumentativos en discursos políticos. Como colofón se incluye también un trabajo que presenta un algoritmo para generar frases homófonas de calambures desde un enfoque fonético. Los juegos de palabras son un tipo de humor verbal ampliamente utilizado en todo el mundo.

Estamos seguros que la lectura de los trabajos incluidos dará una visión muy amplia del estado del arte de la utilización de metodologías inteligentes para soportar el proceso educativo sin limitaciones de tiempos o presencia física.

Carlos Alberto Reyes García
Editor invitado
INAOE, Puebla, México

Septiembre 2016

Table of Contents

	Page
HABCO: Herramienta informática para la automatización de la calificación de exámenes para apoyo al docente	9
<i>Luis Javier Mona Peña</i>	
Diseño de recurso educativo abierto con un sistema experto para la enseñanza-aprendizaje de base de datos	23
<i>Erick Nila Méndez, Cristina Juárez Landín, José Luis Sánchez Ramírez, Anabelem Soberanes Martin</i>	
Aplicación de procesos Markovianos para recomendar acciones pedagógicas óptimas en tutores inteligentes	33
<i>Hermilo Victorio Meza, Manuel Mejía-Lavalle, Alicia Martínez Rebollar, Obdulia Pichardo, Grigori Sidorov</i>	
Propuesta para reducir espacio de búsqueda en entornos virtuales	47
<i>Víctor Tomás Tomás Mariano, Felipe de Jesús Núñez Cárdenas, Efraín Andrade Hernández</i>	
Hacia un modelo de interfaces multimodales adaptables a los canales de aprendizaje en aplicaciones colaborativas como apoyo a la educación	57
<i>Itzel Alessandra Reyes Flores, Carmen Mezura-Godoy, Gabriela Sánchez Morales</i>	
Una base de conocimientos para asistir el aprendizaje de la programación	69
<i>Juan P. Ucán, Antonio A. Aguilera, Raúl A. Aguilar</i>	
Sistema tutor afectivo para el aprendizaje de las matemáticas usando técnicas de gamificación	83
<i>Ramón Zatarain-Cabada, María Lucía Barrón-Estrada, Jorge García-Lizárraga</i>	
PREMOC: Plataforma de reconocimiento multi modal de emociones	97
<i>Ramón Zatarain-Cabada, María Lucía Barrón-Estrada, Gilberto Muñoz-Sandoval</i>	
Sistema tutor afectivo para la enseñanza de lógica algorítmica y programación	111
<i>Ramón Zatarain-Cabada, María Lucía Barrón-Estrada, José Mario Ríos-Félix, Giner Alor-Hernandez</i>	

Evaluando afecto en un entorno de aprendizaje para Java	123
<i>Ramón Zatarain-Cabada, María Lucía Barrón-Estrada, Francisco González-Hernández, Giner Alor-Hernandez</i>	
Sistema de reconocimiento multimodal de emociones relacionadas al aprendizaje en dispositivos móviles	135
<i>María Lucía Barrón-Estrada, Ramón Zatarain-Cabada, Claudia Guadalupe Aispuro-Gallegos</i>	
Una revisión sistemática de técnicas de gamification en aplicaciones educativas inteligentes	149
<i>Humberto Marín Vega, Giner Alor Hernandez, Ramón Zatarain Cabada, M. Lucía Barrón Estrada</i>	
Identificación automática de marcadores argumentativos en discursos políticos	165
<i>Felipe Ojeda-Cruz, Noé Alejandro Castro-Sánchez, Héctor Jiménez-Salazar</i>	
Modelo difuso para evaluar la pertinencia de un programa educativo de nivel superior	177
<i>Sandra Silvia Roblero Aguilar, Héctor Rafael Orozco Aguirre, Saturnino Job Morales Escobar</i>	
Algoritmo para generar frases homófonas de calambures des de un enfoque fonético	191
<i>Roberto Villarejo-Martínez, Noé Alejandro Castro-Sánchez</i>	
Metodología para la creación de objetos de aprendizaje adaptables al estilo de aprendizaje	203
<i>Rodolfo Ibarra-Orozco, Aremy Virrueta-Gordillo, Benedicto Ramírez-Santiago, Flavio Castillo-Silva</i>	

HABCO: Herramienta informática para la automatización de la calificación de exámenes para apoyo al docente

Luis Javier Mona Peña

Instituto Tecnológico de Saltillo, Departamento de Ingeniería Eléctrica–Electrónica,
México

lmona@itsaltillo.edu.mx

Resumen. El desarrollo de software es una herramienta poderosa para auxiliar al hombre en múltiples áreas de su quehacer cotidiano. En áreas de evaluación del nivel cognitivo, de aptitudes, de desempeño entre otros, es especialmente útil ya que existen herramientas de medición diseñadas para que por medio de exámenes de opción múltiple cuantificar dicho nivel cognitivo y demás habilidades y competencias. El presente artículo se enfoca al uso del software LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) en el desarrollo de una herramienta capaz de auxiliar a los profesionales que utilizan dichas pruebas en la evaluación y calificación de las mismas de una manera sencilla, rápida, eficiente, confiable, con rastreabilidad de la información y flexible para cualquier formato de examen de alveolos impreso. El software es flexible en el aspecto de que no importa el formato utilizado (número de alveolos, opciones disponibles de respuesta, distribución geométrica) de las hojas de respuesta de las pruebas, ya que solamente se requiere entrenar una vez al sistema con las respuestas correctas. Las hojas de respuestas de las pruebas son digitalizadas por medio de un escáner para después utilizar algoritmos de procesamiento digital de imágenes (Visión por Computadora) para realizar el análisis con el propósito de verificar respuestas correctas y acto seguido obtener la ponderación de cada pregunta las cuales están en una base de datos de SQL Server. Una vez obtenido resultados estos se exportan a un formato en Microsoft Excel para el posterior manejo de la información obtenida.

Palabras clave: Alveolo, excel, LabVIEW, SQL, visión.

HABCO: Computer Tool for Automatic Exam Evaluation to Help a Teacher

Abstract. Software development is a powerful tool to help in many areas of the men daily work. In areas such as evaluation of cognitive level, skills, performance, among others, it is particularly useful because there are measurement tools designed in order to use multiple-choice tests for measuring cognitive level and other skills and competencies. This article focuses on the use

of software LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) to develop a tool to assist professionals who use these tests in evaluating and rating them in a simple, fast, efficient, and reliable way, with traceability information and flexible for holding any kind of printed multiple choice test format. The software is flexible in the way that doesn't matter the format used (number of alveoli, response options, geometric distribution) of the test's answer sheets, since it is only required to train once the software system with the correct answers. The test's answer sheets are digitized by a scanner and then use algorithms of digital image processing (Machine Vision) for analysis in order to verify correct answers and immediately obtain the weighting of each question which are in a database in SQL Server. After obtaining these results the data are exported to a Microsoft Excel format for further handling of the information obtained.

Keywords: Alveoli, excel, LabVIEW, SQL, vision.

1. Introducción

El uso de la tecnología ha potenciado la aplicación de técnicas y herramientas que permiten obtener información sobre diversos aspectos de la naturaleza y en específico del hombre mismo y su entorno. En el caso del ramo educativo, existen herramientas para evaluar y calificar diferentes áreas de conocimiento, aptitudes y habilidades tales como la prueba ENLACE (Evaluación Nacional del Logro Académico en Centros Escolares) y el examen del CENEVAL (Centro Nacional de Evaluación). Estas pruebas están diseñadas para que el presentante responda a los reactivos por medio de alveolos (opción múltiple), la evaluación y calificación de dichos exámenes se realiza por medio de herramientas de software que permite tener en pocos minutos el resultado de miles de exámenes. En el caso del docente, se tiene libertad sobre el diseño y aplicación de sus propios exámenes, sin embargo la labor de evaluación y calificación se puede tornar desgastante debido a la cantidad de preguntas y el número de alumnos a los que se les aplica la prueba, llegando a dedicar horas enteras a la evaluación y calificación al realizar estas actividades de forma manual por no tener una herramienta que les auxilie en dichas tareas.

El reto principal era desarrollar una herramienta por software altamente confiable y fácil de usar para la evaluación y calificación automática de exámenes de respuestas tipo alveolo que fuera universal, es decir, que no importara la cantidad de preguntas, el número de opciones de respuesta, el formato de distribución, el tamaño de la hoja, etc. Además que fuera capaz de guardar la información en una base de datos en SQL Server los datos del presentante, la ponderación individual de cada reactivo, la calificación final y poder cambiar rápidamente la ponderación de los reactivos modificando los valores en una tabla en la base de datos. El sistema es capaz de detectar si más de una opción por reactivo es seleccionada en el caso de tratar de engañar al sistema y tener tolerancia a los patrones e intensidad del relleno de la opción del reactivo. Por último, realizar el reporte de resultados y exportarlo a formato de Microsoft Excel para que el usuario final realizara las operaciones estadísticas que considere necesarias.

2. Metodología

Para la implementación de la aplicación se definió la siguiente metodología la cual sigue en orden secuencial las etapas de acondicionamiento, obtención, análisis y presentación de la información:

1. **Digitalización de las hojas de respuesta de la pruebas.** Esta etapa es la que alimenta al sistema, requiere que el usuario por medio de un escáner digitalice las pruebas y las guarde en el formato de su preferencia, siendo recomendable los formatos .jpg y .png por la relación de calidad-tamaño de archivo que presentan. Cada archivo se guarda en disco con el nombre del presentante que corresponda y se almacena en una carpeta que se llama “Exámenes” que es una carpeta que el sistema define por default para la búsqueda de las imágenes para su proceso.
2. **Calibración del sistema.** Dentro de calibración de sistema se definen las regiones de búsqueda que serán utilizadas para determinar por medio de algoritmos de procesamiento de imágenes del toolkit de **Machine Vision del software LabVIEW 2015** de National Instruments si las respuestas son correctas o no. La información de las coordenadas de las regiones de búsqueda tanto de intensidad de píxeles como de búsqueda de patrones se guardan en disco duro utilizando un formato de archivo binario para aumentar la velocidad de acceso a la información y la eficiencia en el guardado de la misma.
3. **Conexión con la base de datos en SQL Server Express 2014.** Se define una base de datos relacional con dos tablas. Una de ellas de nombre “Alumnos” tiene diversos campos como Nombre, un campo por reactivo donde se pondrá la ponderación lograda y otro campo con la suma de todas ponderaciones o calificación general por alumno. Una segunda tabla “Ponderaciones” tiene un campo por cada reactivo donde el usuario especifica el valor de ponderación, de esta forma se logra flexibilidad de la aplicación, ya que el usuario puede cambiar a su criterio las ponderaciones de los reactivos en el momento que lo desee sin que tenga que programar nada.
4. **Análisis de las imágenes.** Determinar la cantidad de respuestas correctas según el Master realizado en la etapa de calibración de sistema se utilizan algoritmos de conteo de píxeles e intensidad de nivel de grises necesarios para corroborar si la opción que el presentante seleccionó es la correcta o no. En esta misma etapa se utilizan algoritmos de búsqueda de patrones con herramientas del toolkit de Machine Vision, esta búsqueda de patrones tiene como objetivo determinar que el presentante no haya seleccionado más de una opción por reactivo lo cual podría ser una forma engañar el sistema, por lo tanto si existe más de una respuesta seleccionada el sistema determinará que la respuesta es incorrecta a pesar del resultado del análisis de píxeles.
5. **Actualización de la Información en la base de datos.** La información obtenida por medio de los algoritmos del análisis de imágenes es actualizada en la base de datos dentro de la tabla “Alumnos” donde queda respaldada para su consulta posterior.
6. **Generación del reporte final de calificaciones.** Etapa que genera el reporte final de calificaciones a nivel local y remoto. La información proveniente del

análisis de la información es utilizada para generar una tabla ASCII que se visualiza dentro de la misma interfaz de la aplicación donde cada renglón es el registro del alumno y cada columna la ponderación del reactivo que corresponda. Además por medio de **conectividad ActiveX** se genera un reporte en **Microsoft Excel** para mayor portabilidad de la información generada además de ser una herramienta que la mayoría de los docentes o profesionales utilizan para el análisis estadístico.

3. Implementación de la aplicación.

El software que se utilizó para el desarrollo del sistema de software es National Instruments LabVIEW 2015 ya que integra herramientas de Procesamiento Digital de Imágenes, Conectividad a bases de datos y ActiveX que son las herramientas seleccionadas en la metodología para la implementación del desarrollo de software. Además el programa es bajo el estilo de programación basada en eventos, esto para eficientar el uso de los recursos de hardware y software de la computadora que aloje la aplicación, ya que solo requerirá la atención de ésta cuando el usuario interactúe con el programa sin necesidad de realizar rutinas cíclicas (pulldown) que requieren gran parte de la atención del microprocesador.

El primer paso después de que las imágenes ya están digitalizadas es abrirlas para su análisis. El nombre de cada archivo debe corresponder al nombre del alumno, esto con el propósito de que el software tome el nombre del alumno directamente del archivo y lo guarde en la base datos para mayor comodidad del usuario. El algoritmo toma el nombre del archivo para actualizar la tabla “Alumnos” en el campo “Nombre” de la base de datos y crear los registros con los nombres de los alumnos que corresponde a cada examen. Esto se realiza utilizando comandos SQL. Cada imagen requiere una referencia para su uso la cual es creada por el subvi **IMAQ Create** del toolkit Vision Development System. La figura 1 ilustra el algoritmo completo de creación de referencias a las imágenes, obtención del nombre del archivo y la actualización en el campo “Nombre” de la tabla “Alumnos”.

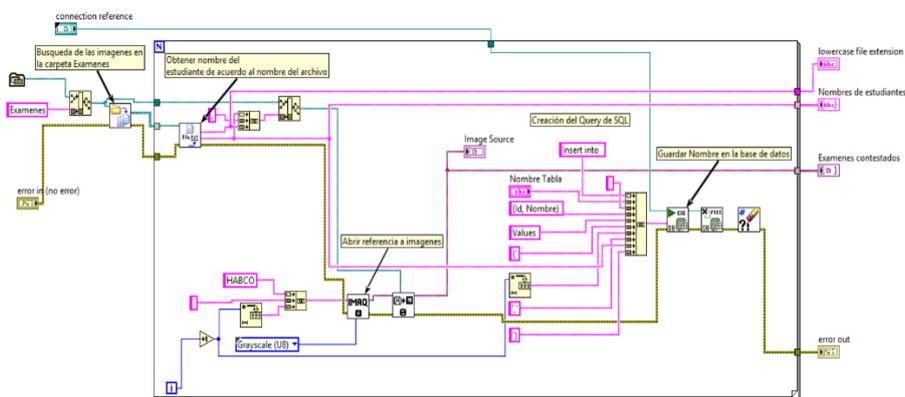


Fig. 1. Obtención de referencias de imágenes y obtención para guardado automático de los nombres de los alumnos.

Este subvi (sub instrumento) da como salida arreglos con la información que requerirán los elementos posteriores para el análisis de las imágenes. El icono de dicho subvi se muestra en la figura 2.

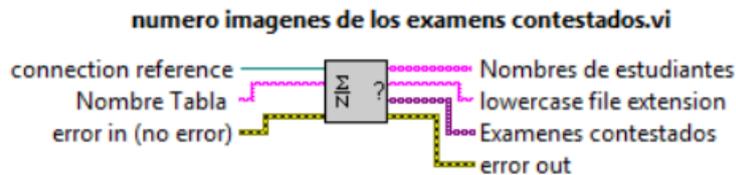


Fig. 2. Icono del Subvi de obtención de referencias de imágenes.

Ya con la referencia de las imágenes, se crean las regiones de búsqueda (ROIS Descriptors) de las respuestas correctas, ver figura 3. Son dos regiones de búsqueda, la primera de ellas se dibuja sobre la opción de la respuesta correcta, esta región es utilizada por el sistema para medir la intensidad de los píxeles buscando la tonalidad negra indicativa que se ha rellenado la opción. La segunda región de búsqueda se utiliza para buscar el patrón de respuesta rellena y por lo tanto obscura, esto se implementa como medida de seguridad necesaria para evitar que se seleccione más de una opción y que el criterio único del conteo de píxeles quede rebasado.

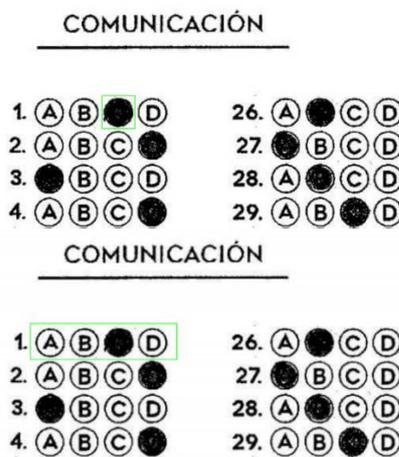


Fig. 3. Determinación de regiones de búsqueda de respuesta correcta y respuesta única.

De esta manera, si más de una opción es seleccionada, no importa si entre ellas va la respuesta correcta, el sistema lo considerará como pregunta mal contestada. La figura 4 muestra el código de implementación de la creación de las regiones de interés que serán utilizadas para la búsqueda de intensidad de píxeles y de patrones.

La información de las coordenadas de las regiones de interés es almacenada en dos arreglos de clusters que indican la descripción del rectángulo de búsqueda. Dicho cluster incluye la coordenada superior, la inferior y la rotación del rectángulo que define la región de interés o de búsqueda. Esta información se guarda en disco duro en formato

binario y sirve como calibración del examen Master para efectuar la evaluación de los exámenes.

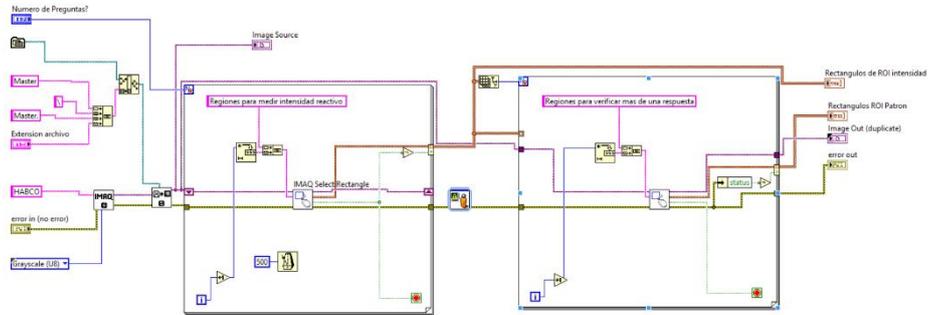


Fig. 4. Código de implementación de regiones de interés del sistema.

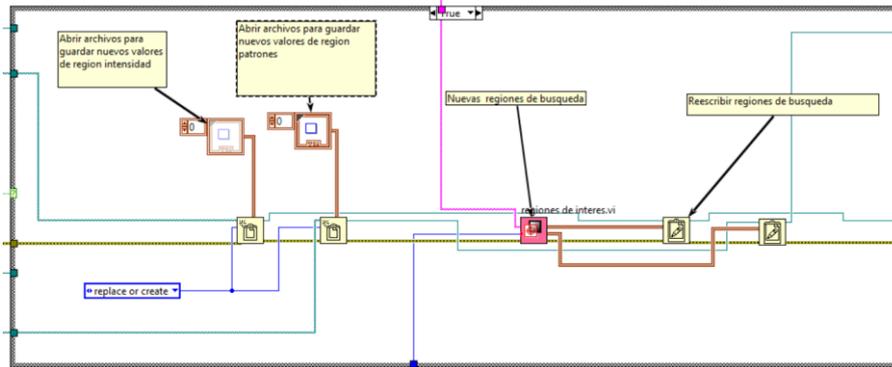


Fig. 5. Código para guardado de información de las regiones de interés en disco duro.

Ya definidas las regiones de interés, el sistema tiene toda la información para realizar la evaluación de los exámenes al buscar en primer lugar la intensidad de los píxeles (niveles de escala de grises) y en segundo término la búsqueda de patrones. La búsqueda de intensidad de píxeles tiene como propósito corroborar que el alumno escogió la respuesta correcta al repintar la opción seleccionada quedando de un color oscuro comparado con el resto de las opciones, tal como se muestra en la figura 3.

Dentro de las herramientas del módulo de visión de LabVIEW el subvi IMAQ Light Meter (1) realiza el conteo de la escala de grises en un rango de 0 a 255, siendo el blanco extremo un valor de 255 indicando la ausencia de pigmentaciones y el negro con un valor de 0 para el negro intenso. La figura 6 indica el código que implementa el subvi IMAQ Ligth Meter.

La salida de este subvi son valores estadísticos de la intensidad de escala de grises, siendo el valor promedio **Mean Value** el parámetro que utiliza el sistema para determinar si el usuario eligió la opción correcta. Utilizando las regiones de interés previamente obtenidas, se evalúa cada imagen (examen) obteniendo el valor promedio en la escala de grises pregunta por pregunta. El resultado es un arreglo bidimensional que contiene la información de cada examen junto con el conjunto de valores de

medición de píxeles de cada pregunta. El código que se muestra en la figura 7 indica el código implementado para realizar la comprobación.

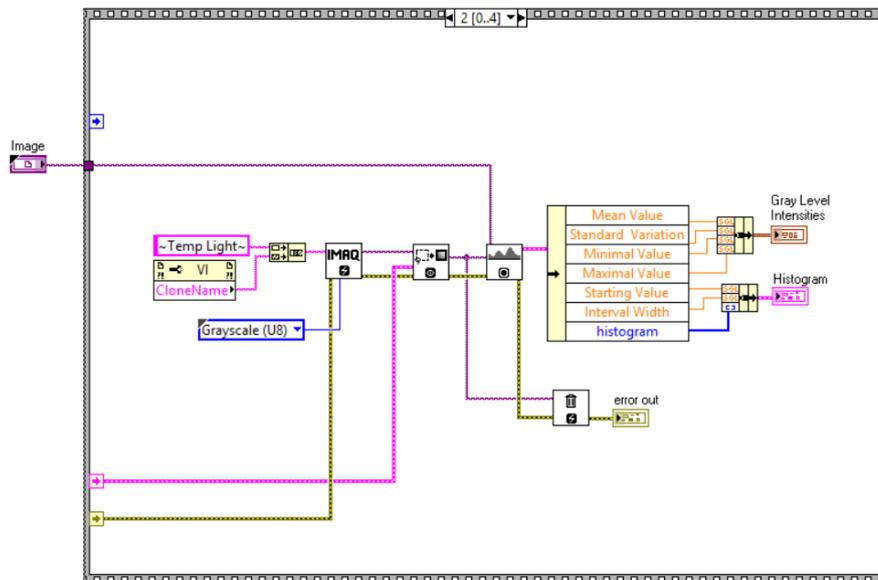


Fig. 6. Código del subvi IMAQ Ligth Meter.

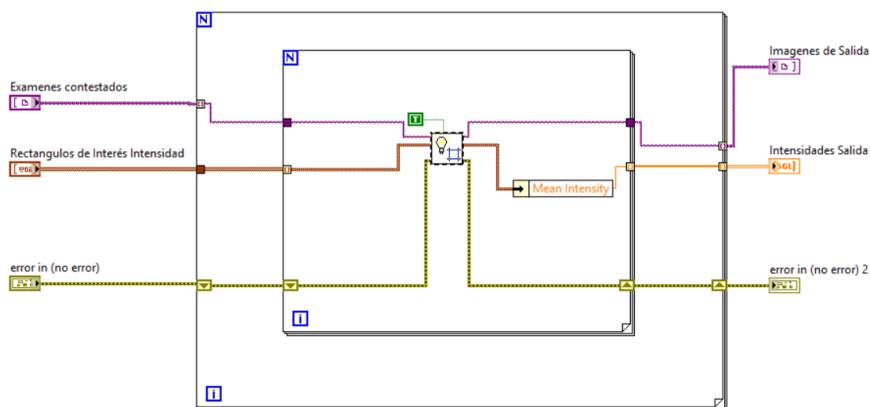


Fig. 7. Obtención de los valores de intensidad de píxeles de cada examen.

Para la comprobación de respuesta única por pregunta del examen se realiza una búsqueda del patrón de respuesta contestada, que sería la opción repintada. Esta acción se realiza con el algoritmo IMAQ Learn Pattern del módulo de visión de LabVIEW (2) donde realiza una búsqueda de una imagen patrón sobre una región de búsqueda en otra imagen, dando como resultado la cantidad de patrones encontrados. Este resultado se

almacena en un arreglo bidimensional que tiene la información de a que examen corresponde cada una de las preguntas junto con el número de patrones encontrados. La cantidad máxima de patrones encontrados por pregunta debe de ser uno, ya que de lo contrario estamos hablando de preguntas con múltiples respuestas seleccionadas. La figura 8 muestra el código implementado para determinar la cantidad de opciones seleccionadas por pregunta de cada examen.

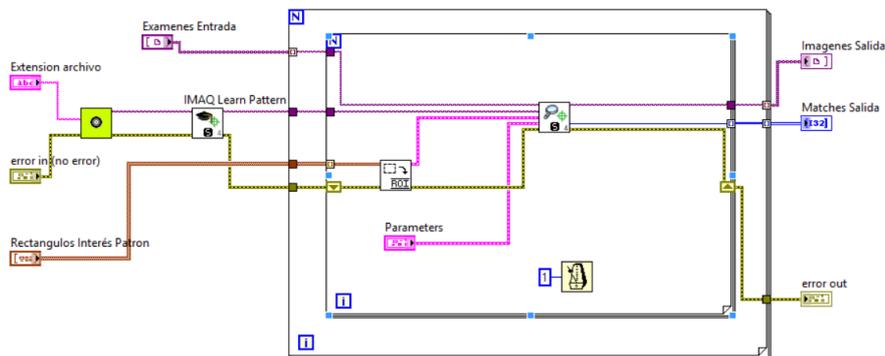


Fig. 8. Código de determinación de respuesta única.

Una vez que el software determina si las respuestas son correctas pasa a la etapa de cuantificación, es decir obtener de la base de datos el puntaje de cada una de las preguntas que colabora para la calificación final. Las ponderaciones de cada respuesta se encuentran en una tabla de nombre “Ponderaciones” de la base de datos de nombre HABCO implementada en SQL Server Express 2014 que utiliza el software, el diseño de la tabla se ilustra en la figura 9.

La imagen muestra la estructura de la tabla 'Ponderaciones' en SQL Server Express 2014. La tabla tiene las siguientes columnas:

Column Name	Data Type	Allow Nulls
Id	int	<input type="checkbox"/>
P1	nvarchar(255)	<input checked="" type="checkbox"/>
P2	nvarchar(255)	<input checked="" type="checkbox"/>
P3	nvarchar(255)	<input checked="" type="checkbox"/>
P4	nvarchar(255)	<input checked="" type="checkbox"/>
P5	nvarchar(255)	<input checked="" type="checkbox"/>
P6	nvarchar(255)	<input checked="" type="checkbox"/>
P7	nvarchar(255)	<input checked="" type="checkbox"/>
P8	nvarchar(255)	<input checked="" type="checkbox"/>
P9	nvarchar(255)	<input checked="" type="checkbox"/>
P10	nvarchar(255)	<input checked="" type="checkbox"/>
P11	nvarchar(255)	<input checked="" type="checkbox"/>
P12	nvarchar(255)	<input checked="" type="checkbox"/>
P13	nvarchar(255)	<input checked="" type="checkbox"/>
P14	nvarchar(255)	<input checked="" type="checkbox"/>
P15	nvarchar(255)	<input checked="" type="checkbox"/>
P16	nvarchar(255)	<input checked="" type="checkbox"/>
P17	nvarchar(255)	<input checked="" type="checkbox"/>

Fig. 9. Vista de diseño en SQL Server Express 2014 de la tabla ponderaciones.

En esta tabla el usuario puede cambiar las ponderaciones de cada pregunta para cuestiones de actualización de las pruebas.

El código que implementa la búsqueda de la tabla “Ponderaciones” utiliza enunciados en SQL para buscar el valor de ponderación de cada reactivo.

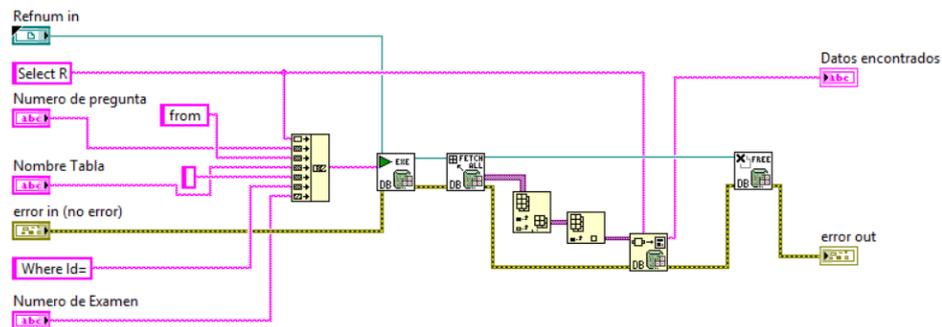


Fig. 10. Código para búsqueda de la ponderación de cada reactivo.

Ya con los valores de ponderación el sistema procede a actualizar cada campo que representa una pregunta del examen (Figura 11), dando el valor de cero cuando la respuesta es incorrecta y el valor correspondiente al valor de ponderación que se leyó de la tabla “Ponderaciones” cuando la respuesta es correcta. La herramienta del toolkit de conectividad a bases de datos **Execute Query (3)** permite ejecutar comandos SQL para la interacción con la base de datos.

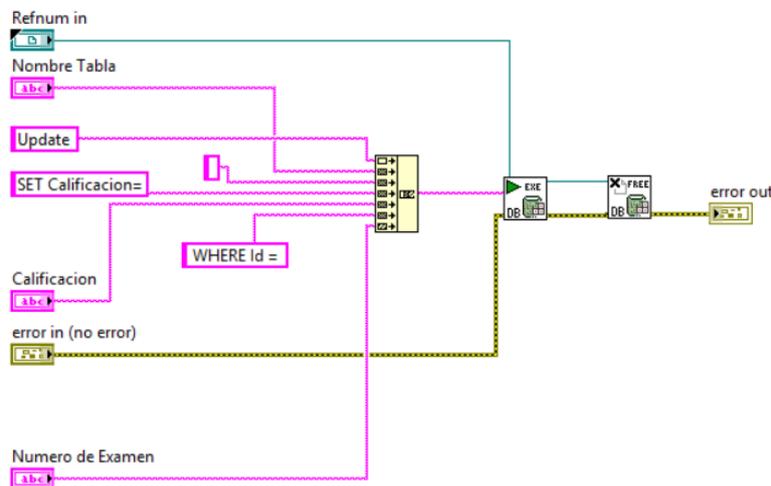


Fig. 11. Actualización de los campos de cada pregunta en la Tabla Alumnos en SQL Server Express 2014.

De la misma manera por medio de comandos en SQL (4) se realiza la tarea de actualizar campos de calificaciones, los cuales son los de la tabla “Alumnos” donde quedarán los registros con las calificaciones de cada examen teniendo como

identificador del registro el nombre del alumno. Un ejemplo de cómo queda la tabla Alumnos de la base de datos del sistema se muestra en la figura 12 en su interfaz de SQL Server.

Id	Nombre	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	
1	Ana Lucía Martínez Reta	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	
2	Ana Paula Larrazábal López	0	0	6	12	12	9	0	0	0	9	0	0	0	0	12	0
3	Carlos Alberto Gómez Hernández	6	9	0	0	12	9	0	6	6	0	18	0	0	0	12	12
4	Dulce María Treviño Leal	0	9	0	0	12	9	6	0	6	0	0	0	0	0	0	0
5	José Diboldox Hassaf	0	9	6	12	12	0	0	0	6	0	0	0	0	0	0	0
6	José Emiliano Moncada Quintanilla	0	0	0	0	0	9	0	0	0	9	0	0	0	0	12	12
7	Liza Regina Comejo Valdez	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	12
8	Marian Alejandra Luna Hernández	0	0	0	0	0	9	6	0	0	0	0	0	0	0	0	0
10	Miniam Cabello Vallarta	0	9	0	0	0	0	6	0	6	9	0	0	12	0	0	0
11	Roberto Sosa Dávila	0	0	0	0	0	9	6	0	6	9	0	0	0	0	0	0
12	Ximena Sofía Morales Padron	6	9	6	12	12	9	6	6	6	9	0	0	12	0	0	0
13	Luis Adrián Dogart de la Paz	6	0	0	0	12	9	0	0	6	9	18	18	0	12	0	0
14	Mariana Sánchez Mulloz	0	0	0	0	0	0	0	0	0	9	0	0	12	12	12	0
15	Scarlett Lara Valenzuela	0	9	6	12	12	9	0	0	0	9	0	0	0	12	12	0
16	Ximena Villarreal Valdez	0	9	6	12	12	0	0	0	6	0	0	0	0	12	0	0
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Fig. 12. Tabla Alumnos actualizada en SQL Server Express desde LabVIEW.

La calificación final es el resultado de la suma de todas las ponderaciones de los reactivos, dicha información se realiza con el siguiente código que involucra el promedio de los elementos del arreglo de calificaciones y la actualización en el campo “Calificación Final” de la tabla Alumnos. La figura 13 indica el código implementado.

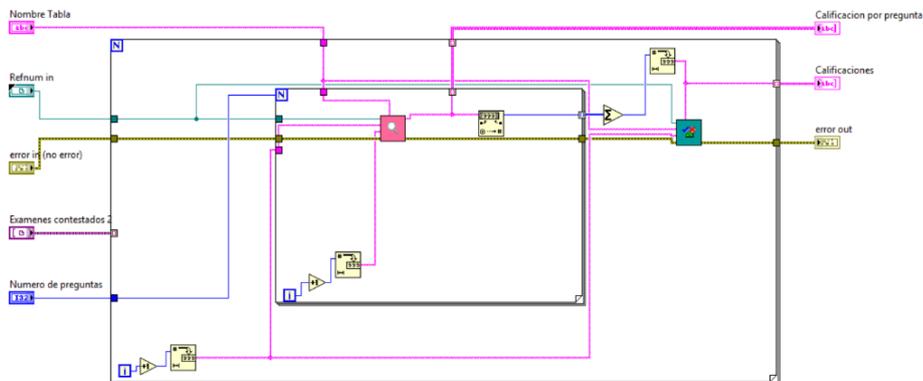


Fig. 13. Código para la obtención de la calificación final.

Finalmente la creación del reporte en Microsoft Excel se implementó utilizando el toolkit de Report Generation (5) de LabVIEW que a través de referencias ActiveX logra la comunicación y conectividad requerida. Dentro de LabVIEW se personalizó el tipo de reporte que incluye toda la información que se encuentra en la tabla Alumnos de la base de datos, es decir, Nombre del alumno, calificación por pregunta y calificación final. La Figura 14 ejemplifica parte del código implementado para la creación del reporte final de calificaciones del examen. El cual queda listo para su posterior análisis.

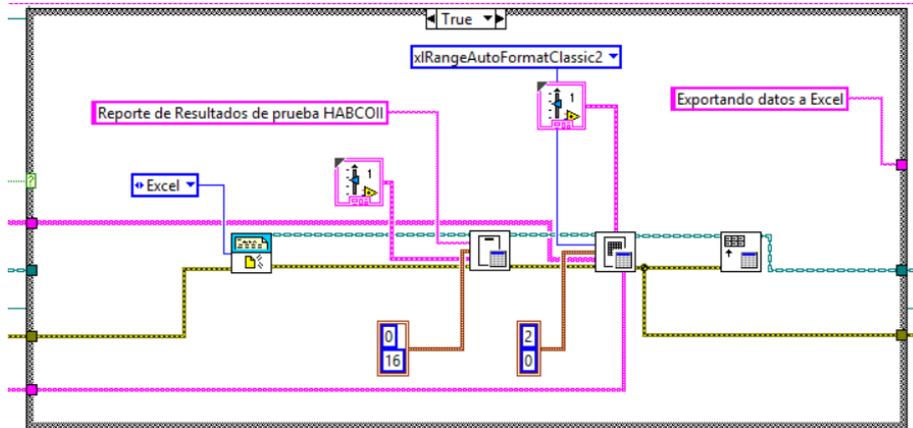
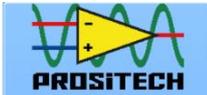


Fig. 14. Creación del reporte de calificación final.

4. Resultados

El resultado es un software que de manera amigable permite al usuario la calificación automática de sus pruebas de alveolos en un tiempo de aproximadamente un minuto para una cantidad de setenta y cinco exámenes, ahorrando una gran cantidad de tiempo y desgaste del docente, además de eliminar el error humano.



PROSITECH

Software para la Evaluación y Obtención de
Resultados de la prueba HABCO



HABCO

Numero de preguntas del Examen: 100

Reentrenar Regiones de búsqueda

EVALUAR

Obtener Puntuaciones

Exportar Datos a Excel

Stop

COMUNIC

1. (A) (B) (C) (D)

2. (A) (B) (C) (D)

3. (A) (B) (C) (D)

4. (A) (B) (C) (D)

5. (A) (B) (C) (D)

6. (A) (B) (C) (D)

7. (A) (B) (C) (D)

Nombre	R1	R2	R3	R4	R5
Alejandra Garcia Gómez	0	0	0	0	0
Alejandro Jr Gómez Vázquez	0	0	6	12	12
Ana Camilla Torres Mac Greg	6	9	0	0	12
Ana Gabriela Sánchez Gonzal	0	9	0	0	12
Ana Paula Antope Revilla	0	9	6	12	12
AnaLucia Leal Ballesteros	0	0	0	0	0
Andrea Verónica Valero Mart	0	0	0	0	0
Cecilia Villegas Robles	0	0	0	0	0
David Alejandro Flores Meza	0	0	0	0	0
Eduardo Locaira Mata	0	9	0	0	0
Emilia Leal Ballesteros	0	0	0	0	0
JoseEmilio Moncada Quintar	6	9	6	12	12
Luis Adrián Dogart de la Paz	6	0	0	0	12
Luis Diego Barajas Lozano	0	0	0	0	0
Mariany Alejandra Luna Henri	0	0	0	0	0
Mariana Sánchez Muñoz	0	0	0	0	0
Melany Hatziy Leal Rios	0	0	0	0	0
Mónica Uribe Aguirre	0	0	0	0	0
Scarlett Lara Valenzuela	0	9	6	12	12
Ximena Villarreal Valdez	0	9	6	12	12

Fig. 15. Interfaz de Usuario.

La flexibilidad del mismo permite su uso para cualquier formato de examen, donde solo requiere la calibración inicial para definir las regiones de interés o de búsqueda de

respuestas correctas y la definición del patrón de búsqueda de respuesta contestada. No importa el formato de imagen con que se guarda después de la digitalización de cada prueba, ya que el sistema automáticamente cambia el formato de la imagen a 8 bits a blanco y negro para eficientar el uso de los algoritmos de visión utilizados. La interfaz de usuario resultante se muestra en la figura 15.

La etapa de prueba y validación del software se está haciendo en la empresa Reingeniería Didáctica S. de R.L de C.V. y en la empresa Prositech Tecnología Aplicada S.A. de C.V. donde a la fecha se han calificado más de 400 exámenes de valoración de aptitudes de comunicación y habilidad matemática denominado HABCOII diseñado por expertos en pedagogía de Reingeniería Didáctica que sirve como examen de admisión y ubicación en el Colegio Americano de Saltillo (COLAM). La figura 16 muestra un ejemplo de la generación de reporte que arroja el software de los datos generados. Estos resultados son posteriormente interpretados para obtener la información que se requiera.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	
1	Reporte de Resultados de prueba HABCOII																																									
2																																										
3	Nombre	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25	R26	R27	R28	R29	R30	R31	R32	R33	R34	R35	R36	R37	R38	R39	R40	
4	Adriana Jimenez Soto	0	9	6	0	12	9	0	0	6	0	18	18	0	12	0	3	3	3	3	0	12	12	12	12	9	0	0	0	0	0	15	12	9	0	9	18	6	6	9		
5	Adriana Lisette Aguirre de la Rosa	6	9	6	12	12	9	6	0	0	9	0	0	12	0	3	3	3	3	3	12	12	12	12	12	9	6	15	0	0	9	15	12	0	9	9	18	6	6	9		
6	Adrián Moller Dávila	0	0	6	12	12	0	6	0	6	9	18	18	0	12	12	3	3	3	3	3	12	12	12	12	12	9	6	15	0	18	9	0	0	9	0	9	0	6	6	9	
7	Alba Luna Adsara	0	9	6	12	12	0	6	0	6	9	18	18	12	12	12	3	3	3	3	3	12	12	12	12	12	9	6	15	9	18	0	0	12	0	9	9	0	6	6	9	
8	Alberto Aritpe Alcantara	0	0	6	12	12	9	0	0	6	9	18	0	12	12	12	3	3	3	3	3	12	0	0	12	12	9	6	15	9	0	0	0	12	9	9	9	0	6	6	9	
9	Alejandra Castro Torres	0	0	6	12	12	0	6	0	6	0	0	0	12	12	12	3	3	3	3	3	12	0	0	12	12	9	6	0	0	0	0	12	0	9	9	18	6	6	9		
10	Alejandra Romero Nagafuchi	6	9	6	0	0	0	6	6	6	9	18	18	0	12	0	3	3	3	3	3	12	12	12	12	12	9	0	15	0	18	0	15	12	0	9	9	18	6	6	9	
11	Alejandra Villarreal González	6	9	6	0	12	9	0	0	6	9	18	18	0	12	12	3	3	3	3	3	12	12	12	12	12	9	6	15	0	18	0	15	12	9	9	9	18	6	6	9	
12	Alejandro Charles Oyervides	6	0	6	0	12	0	6	0	0	9	18	18	0	12	0	3	3	0	0	3	12	12	12	0	0	0	15	0	18	0	0	12	9	0	9	0	6	6	9		
13	Alejandro Moreno Noemi	0	0	6	12	12	9	0	0	6	0	18	0	12	0	3	3	3	3	3	0	12	12	12	0	0	0	15	9	0	9	0	0	0	0	9	0	0	0	0	0	
14	Alexis Lizarraga Barrera	6	9	6	0	12	9	6	0	0	9	18	18	12	12	12	0	3	3	3	3	12	0	12	12	0	0	6	15	0	18	0	15	12	9	9	9	18	6	6	9	
15	Ana Pamela Montemayor Pérez	6	0	6	12	12	0	6	0	9	0	0	12	12	12	0	3	3	0	3	3	12	12	12	12	12	9	6	15	0	0	15	0	9	0	9	0	6	6	9		
16	Ana Paula Moreno Medina	0	0	6	12	12	9	0	6	6	9	0	0	12	12	3	3	3	3	3	3	12	12	12	12	12	9	6	15	0	18	0	0	0	0	0	9	18	6	6	9	
17	Ana Sofía González Rodríguez	0	0	0	12	12	0	0	0	6	9	0	0	12	12	3	3	3	3	3	3	12	0	12	0	0	0	15	0	0	0	0	0	0	0	9	9	18	6	6	0	
18	Andrés Salinas Espinosa	6	9	6	12	12	0	0	0	6	0	18	18	12	12	12	0	3	3	3	3	12	12	12	12	12	0	6	15	0	18	0	15	12	0	9	0	9	18	6	0	9
19	Angela Orozco Valdes	6	9	0	12	12	9	0	0	6	0	18	18	0	12	12	3	3	3	3	3	12	12	12	12	12	9	6	0	18	0	0	12	0	9	9	18	6	6	9		
20	Anna Patricia Ibarra	0	9	6	12	12	9	6	0	0	9	0	12	12	0	3	3	3	3	3	3	12	0	0	12	9	6	15	0	18	0	0	0	0	9	0	6	0	6	9		
21	Antonio Ramos Fariás	6	9	6	12	12	9	0	0	6	0	18	18	0	12	3	3	3	3	3	3	12	12	12	12	12	9	6	0	0	0	15	12	0	0	9	18	0	0	9		
22	Daniel Jiménez Botello	6	0	6	12	12	9	6	0	0	9	18	18	12	12	12	3	3	3	3	3	12	0	12	12	0	9	6	0	0	0	15	12	9	9	9	18	6	0	9		

Fig. 16. Reporte en Microsoft Excel generado por el software.

5. Conclusiones

La implementación del desarrollo de Software para la solución de problemas es una herramienta poderosa que contribuye a la eficiencia de procesos, ya que al automatizar integrando diferentes tecnologías tal como conectividad ActiveX, algoritmos de análisis de imágenes y señales, contenedores .NET, protocolos de comunicación entre otros, permite mejorar tiempos de producción y control de calidad.

Se desarrolló una herramienta de bajo costo que no requiere hardware especial para su funcionamiento, ya que funciona en cualquier PC bajo ambiente Windows y los exámenes pueden ser digitalizados con cualquier escáner, esto permite al usuario una mejor planificación de su tiempo al contar con una herramienta auxiliar en una labor fundamental como lo es la evaluación y calificación de pruebas académicas, psicométricas o de cualquier tipo que involucre el formato de opción múltiple por medio de alveolos.

Referencias

1. National Instruments: NI IMAQ Documentation. June Edition. http://zone.ni.com/reference/en-X/help/370281P01/imaqvision/imaq_light_meter__rectangle/ (2015)
2. National Instruments: NI IMAQ Documentation. Edition. http://zone.ni.com/reference/en-XX/help/370281U-01/imaqvision/imaq_learn_pattern_4/ (2015)
3. National Instruments: LabVIEW 2015 Connectivity User Manual. Edition (2015)
4. Microsoft SQL Server Express MSDN <https://msdn.microsoft.com/enus/sqlserver2014express.aspx> (2015)
5. National Instruments: LabVIEW 2015 User Manual. June, Edition (2015)

Diseño de recurso educativo abierto con un sistema experto para la enseñanza-aprendizaje de base de datos

Erick Nila Méndez, Cristina Juárez Landín, José Luis Sánchez Ramírez,
Anabelem Soberanes Martin

Universidad Autónoma del Estado de México, Maestría en Ciencias de la
Computación, C. U. Valle de Chalco, Estado de México,
México

li.ericknm@live.com.mx, cjuarezl@uaemex.mx, jluissar@gmail.com,
belemsobranes@yahoo.com.mx

Resumen. En este artículo se presenta el desarrollo de Recursos Educativos Abiertos (REA) por medio de un Sistema Experto (SE) como complemento al proceso de enseñanza-aprendizaje, y así cumplir con los requisitos principales para facilitar su adición a repositorios de REA. El proceso de desarrollo se basa en el Modelo tecnológico para la creación de un REA propuesto en 2014. Dichos recursos apoyan el aprendizaje y la práctica docente aumentando el acervo de conocimientos y recursos disponibles de forma libre a la comunidad estudiantil. Del mismo modo se extiende la labor del docente fuera del aula complementando y apoyando el aprendizaje del alumno de dos formas, la primera con un REA que cumple con todos los requisitos estandarizados de desarrollo y la segunda con un Sistema Experto que interactúa con el estudiante como si estuviera con el docente de forma presencial evaluando su aprendizaje.

Palabras clave: Recurso educativo abierto, sistema experto, repositorio, enseñanza, aprendizaje, inteligencia artificial.

Design of Open Educational Resource with an Expert System for Teaching-Learning of Data Bases

Abstract. This article describes the development of Open Educational Resources (OER) through an expert system (SE) to supplement the teaching-learning process, and thus fulfill the main requirements for easy addition to OER repositories. The development process is based on the technological model for the creation of OER proposed in 2014. These resources support learning and teaching practice increasing the wealth of knowledge and resources freely available to the student community. Similarly the work of teachers outside the classroom extends complementing and supporting student learning in two ways, first with an OER that meets all standard requirements of development and the second with an expert system that interacts with the student as if I were with the teacher in person evaluating their learning.

Keywords: Open educational resource, expert system, repository, teaching, learning, artificial intelligence.

1. Introducción

La UNESCO [1] menciona que el acceso universal a la educación de calidad contribuye al desarrollo sostenible de la sociedad y de la economía.

El rol de la tecnología en su rápido y cambiante crecimiento demanda respuestas ágiles en el campo educativo, tanto por parte de las instituciones como del personal docente, en la medida en que se implementan tecnologías educativas que enriquecen los ambientes de aprendizaje en el aula [2].

El aumento del acceso a Internet ha promovido aún más el estudio individualizado y en conjunto con las redes sociales y el aprendizaje colaborativo ha generado oportunidades para la innovación pedagógica.

Una gran cantidad de recursos educativos, producto del avance tecnológico, están ahora al alcance de las aulas. Además, a esta evolución se ha sumado una nueva tendencia hacia la apertura y la democratización del conocimiento, y ha dado lugar al movimiento de los recursos abiertos. Con esto surgen como una innovación en la educación los Recursos Educativos Abiertos (REA) [3].

Es de suma importancia la creación e implementación de los REA para impulsar el libre aprendizaje de los alumnos como también apoyar la práctica docente. Sin embargo hay que considerar que no todos los REA existentes son o fueron creados bajo un estándar, además que se considera de manera manifiesta la presencia de los docentes y en consecuencia los alumnos reducen su opción de estudio más allá del aula.

Por lo anterior, es un motivo que impulsa esta investigación y que en conjunto con un Sistema Experto (SE) que complementa al REA para obtener recursos computacionales para la enseñanza y aprendizaje contribuyendo a la línea de Inteligencia Artificial en la educación.

El Sistema Experto hace que el docente este de manera virtual complementando la enseñanza-aprendizaje más allá del aula por lo que el alumno tiene una retroalimentación del tema.

1.1. Recursos Educativos Abiertos (REA)

Los REA son materiales de enseñanza, aprendizaje e investigación, en este caso en soporte digital que son de dominio público o con una licencia abierta lo que permite el acceso gratuito a estos materiales los cuales pueden ser usados, adaptados y redistribuidos.

En 2001, el MIT (Massachusetts Institute of Technology) anunció la publicación de sus cursos en Internet para que fueran accesibles a todo el público, lo que marco el precedente para que la UNESCO en el 2002, en el primer foro mundial sobre recursos educativos de libre acceso definiera el término “recursos educativos de libre acceso” [1].

A partir de ese momento todas las instituciones educativas deben de contar con un repositorio con los recursos educativos que producen, sin embargo de acuerdo a la infraestructura con que cuentan, muchas de ellas no pueden crear sus propios repositorios por lo cual los repositorios de recursos educativos que existen actualmente permiten la adición de REA a instituciones o público en general siempre y cuando cumplan con los estándares establecidos.

Actualmente podemos encontrar repositorios de REA como: TEMOA [4] del Tecnológico de Monterrey, CLARISE [5] formado por instituciones educativas de nivel superior en Latinoamérica y RIUAMEX [6] propio de la Universidad Autónoma del Estado de México por citar algunos que ofrecen REA con información que posee autenticidad, validez y confiabilidad en diversas ramas del conocimiento.

1.2. Sistemas Expertos

La Inteligencia Artificial (IA) es una ciencia reciente en comparación con otras cuyo nombre se acuñó en 1956 por Jonh McCarty. Abarca una gran cantidad de subcampos como la planificación autónoma, juegos, control autónomo, diagnóstico, planificación logística, robótica, procesamiento de lenguaje y resolución de problemas, educación, enseñanza, sistemas tutores y sistemas expertos.

Los SE o sistemas basados en conocimiento, almacenan el conocimiento de expertos en un campo determinado. Para Peña [7] es un Sistema computacional que adquiere conocimiento especializado en un campo específico para explotarlo mediante métodos de razonamiento que emulan el desempeño del experto humano en la solución de problemas.

También puede definirse como un Sistema informático (Hardware y Software) que simula a los expertos humanos en un área de especialización dada [8].

Tabla 1. Clasificación general de Sistemas Expertos.

Por su configuración	Por la tarea que ejecutan
Sistemas Expertos Independientes.	Sistemas Expertos de control.
Sistemas Expertos híbridos.	Sistemas Expertos de formación.
Sistemas Expertos dedicados.	Sistemas Expertos para diagnóstico, depuración y reparación.
Sistemas Expertos que presentan una arquitectura integrada.	Sistemas Expertos para supervisión.
Sistemas Expertos embebidos.	Sistemas Expertos de planificación.
	Sistemas Expertos de diseño.
	Sistemas Expertos de pronóstico y predicción.
	Sistemas Expertos de interpretación.

Giarratano & Riley mencionan que un Sistema Experto consta de [9]:

- **Interfaz de usuario.** Permite la comunicación entre el usuario y el Sistema experto.
- **Medio de explicación.** Explica al usuario el razonamiento del Sistema.
- **Memoria activa.** Contiene los hechos usados por las reglas.
- **Mecanismo de inferencia.** Infiere cuales reglas satisfacen los hechos u objetos ejecutando la prioridad más elevada de las reglas.
- **Agenda.** Creada por el mecanismo de inferencia y contiene una lista con prioridades asignadas a las reglas.

- **Medio para la adquisición de conocimiento (opcional).** Permite introducir nuevo conocimiento al sistema por parte del usuario de manera automática.

Los SE se pueden clasificar de manera general como se muestra en la Tabla 1.

Pero ¿por qué usar sistemas expertos?, al dar respuesta a este cuestionamiento se identifica que los objetivos son variados ya que en el contexto sería difícil imaginar las necesidades para hacer uso de estos sistemas, sin embargo se pueden mencionar los siguientes:

1. Hay pocos expertos humanos y personal con poca experiencia, además de que resulta costosa la capacitación o es de un largo tiempo de preparación.
2. La combinación de conocimientos de varios expertos humanos daría como resultado sistemas expertos más confiables.
3. La agilidad de respuesta es rápida en problemas complicados.
4. El error humano está presente en los diversos problemas complejos por lo que la solución no puede ser fiable.

Dentro de los SE que más renombre han tenido desde su creación se puede mencionar [10]:

- MYCYN. Desarrollado en la Universidad de Stanford se encarga del diagnóstico de infecciones en la sangre y su terapia.
- MACSYMA. Trabaja con cálculo diferencial e integral.
- PROSPECTOR. Prospección y evaluación de yacimientos de minerales principalmente cobre y aluminio.
- DENDRAL. Analiza la estructura molecular de un compuesto.
- XCON. Primero en ser usado comercialmente para la configuración de los sistemas DEC/VAX.

2. Trabajos relacionados

La aplicación de los SE en la educación como se menciona en [11] puede hacer referencia a la aplicación de SE para el diseño de modelos instruccionales de enseñanza enfocado principalmente para el instructor que va a impartir el contenido y la otra aplicación enfocado al aprendizaje asistido por computadora como una herramienta que ayude al proceso enseñanza-aprendizaje de cualquier área del saber y que está enfocado a los alumnos quienes son los usuarios principales del SE.

Se han construido diversos SE que como se menciona en [12] los cuales la mayoría de ellos son prototipos. Se puede citar a SOPHIE y sus respectivas versiones SOPHIE II y SOPHIE III creado por la compañía Bolt, Beranek & Newman que se diseñó para ejercitar la detección y reparación de averías en circuitos electrónicos. SCHOLAR es otro ejemplo, fue uno de los primeros sistemas que modeló el conocimiento procedural en forma de red semántica.

Actualmente existen SE de uso comercial [13] que se utilizan como herramientas para hallar resultados, programar o investigar con la simulación matemática. Los sistemas más conocidos tenemos Mathematica, Matlab y Geometer's Sketchpad.

Mathematica [14] es un sistema que permite aprender, hacer y aplicar la matemática en diversas áreas del conocimiento por medio de cálculos en forma simbólica o numérica y de este modo resolver problemas con cualquier grado de dificultad.

Matlab es una herramienta de cálculo simbólico ofreciendo la funcionalidad de una súper calculadora y un intérprete de un lenguaje de programación propio por lo que es utilizado para el aprendizaje automático.

Geometer's Sketchpad [13] fue diseñado principalmente para servir como herramienta en el proceso de enseñanza-aprendizaje, es producido por Key Curriculum Press y fue desarrollado para la enseñanza de la geometría en la educación secundaria, en la práctica se ha visto que es una herramienta pedagógica que se puede utilizar en niveles superiores. Permite diseñar experiencias de aprendizaje en áreas como: la teoría de funciones, cálculo y ecuaciones diferenciales.

Existen otros tipos de SE que sólo son proyectos de investigación académico y tienen un objetivo particular. Se puede mencionar AgentGeom [15] desarrollado por Pedro Cobo y Josep M. Fortuny que es un Sistema Tutorial Inteligente que cuenta con una base de conocimientos definido por sus autores "espacio básico de un problema" que contiene todas las formas posibles de resolver un problema por parte de un experto. El sistema recibe las acciones por parte del alumno que son el dibujo de gráficas y las deducciones para la posible solución de un problema y al mismo tiempo el sistema le va indicando al alumno si lo que está realizando está mal por medio de mensajes de error por lo que el objetivo del sistema es ayudar directamente al alumno en la resolución del problema. La interfaz dedicada para el profesor permite asignar los problemas a sus alumnos o crear problemas nuevos.

3. Materiales y métodos

El diseño de los REA se basa en una metodología que indica los pasos a seguir en este caso el Modelo tecnológico para la creación de un Recurso Educativo Abierto (REA) como dice Ramírez [16] apoya como guía para la solución de problemas y se utiliza como auxiliar en las decisiones y la elaboración del diseño instruccional. Aquí se menciona la parte tecnológica, es necesario interactuar con la tecnología. Siguiendo los pasos de la metodología y el análisis de diferentes diseños instruccionales se puede realizar el diseño instruccional para la creación de los REA. El Modelo tecnológico consta de 5 etapas que se describen brevemente a continuación:

- **Análisis.** Planeación o secuencia didáctica.
- **Diseño.** Prototipo, evaluaciones del temario y evaluaciones del REA.
- **Producción.** Verificación de acuerdo a la planeación y al prototipo del diseño, consulta del catálogo de software multimedia, sugerencias de páginas de música o imágenes abiertas, decisión e inicio de la parte multimedia, realizar cambios de acuerdo a la sugerencia del docente.
- **Desarrollo.** Integración de prototipos, consulta de catálogo de software para realizar el REA, se documenta cada parte del REA, consultar y establecer metadatos.
- **Implementación.** Autoría, consultar catálogo de licenciamiento creative commons, validación, retroalimentación.

Al ser un recurso compartido y de uso libre se trabaja con herramientas Open Source. Para el caso del diseño del prototipo se utiliza Evolus Pencil [17], para el desarrollo del REA como recurso Web se utiliza la herramienta eXeLearning [18] que es una herramienta de autor que sirve de apoyo a los docentes en la creación de contenidos Web, para las pruebas se utiliza Google Drive como servidor alojando el recurso.

Los SE de manera general están compuestos de los elementos que se muestran en la arquitectura de la Fig. 1.



Fig. 1. Elementos de un sistema experto [9].

El diseño de un Sistema Experto se puede guiar de la arquitectura general dependiendo de la funcionalidad que se requiera.

También se puede hacer uso de alguna metodología para el diseño de sistemas, sin embargo se considera para este trabajo usar el Modelo lineal del ciclo de vida para desarrollar Sistemas Expertos, este modelo permite que solo se lleven a cabo aquellas tareas necesarias en cada etapa, por lo tanto no es obligatorio cubrir todas las tareas marcadas. La Fig. 2 muestra los pasos de esta metodología.

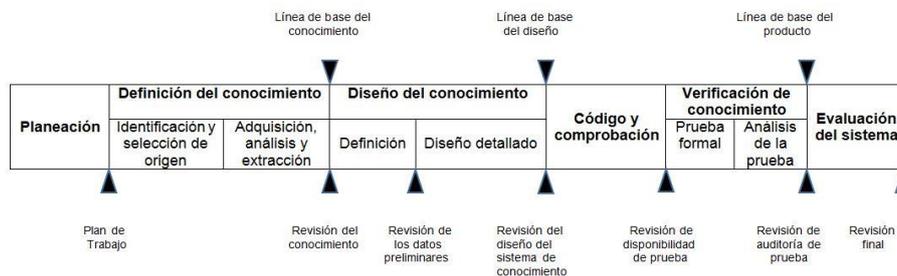


Fig. 2. Etapas del Modelo lineal del ciclo de vida para desarrollar sistemas expertos [9].

En la codificación del SE se hace uso de un lenguaje de programación lógico y siguiendo la línea del uso de herramientas Open Source se utiliza el programa SWI-Prolog [19] que permite crear sistemas basados en reglas y hechos.

En cuanto a la etapa de Definición del conocimiento, el origen del conocimiento se hace a través de libros, revistas, páginas de Internet, que se considera fueron creados por expertos, sin embargo, se pide el apoyo de un docente experto que imparte la materia de Base de Datos en el Nivel Medio Superior para presentar y explicar de una manera fácil y sencilla a los alumnos el contenido del REA y de las preguntas que se encuentran en el SE.

4. Resultados

El REA propuesto está diseñado para la enseñanza de los Fundamentos de bases de datos en el Nivel Medio Superior basado en el plan de estudios del bachillerato de la UAEM de la materia Informática Administrativa [20]. Dicho REA cuenta con autoevaluaciones al término de cada tema, además se pretende que al finalizar todo el curso el alumno pueda acceder a una evaluación final, donde esta fase del trabajo es a través de un SE. Está desarrollado en eXeLearning por lo que se muestra como una página web que cuenta con un menú mostrando todos los temas del módulo, referencias y material de apoyo, es en este último apartado donde se encuentra como archivo a descargar el SE el cual se muestra como archivo setup conteniendo todo lo necesario para instalarse en los equipos que utilicen los alumnos sin necesidad de tener instalado SWI-Prolog para poder trabajarlo. El SE se instala siguiendo un típico asistente de instalación y queda agregado como un programa más para su uso.



Fig. 3. Etapas del SE con el Modelo lineal del ciclo de vida para desarrollar sistemas expertos.

El SE propuesto se encuentra en fase de prototipo y está diseñado para el REA antes mencionado y se considera de tipo formación, ya que posee un base de conocimientos

que se empleara en el adiestramiento de los alumnos y poder detectar las lagunas de conocimiento que el alumno padezca como conceptos o ideas equivocadas asimiladas con anterioridad. Por tal motivo, se le da al alumno una retroalimentación de lo visto en el recurso a través de una serie de preguntas. El modelado del SE está basado en el Modelo lineal del ciclo de vida para desarrollar sistemas expertos, en la Fig. 3 se puede observar brevemente lo realizado en cada una de las etapas del modelo en el SE.

Como se mencionó en el punto anterior la retroalimentación que ofrece el SE es con una serie de preguntas clasificadas por los temas principales que contiene el REA, que en este caso son 3: Introducción, modelo de datos y bases de datos relacionales que se definen dentro del código del Sistema como reglas Fig. 4.

```
tema (introduccion) :-introduccion,!.
tema (modelo_de_datos) :-modelo_de_datos,!.
tema (bdrelacionales) :-bdrelacionales,!.
tema ('Indefinido').
```

Fig. 4. Reglas establecidas en el SE para la selección del tema a reparar.

Cuando se ejecuta el SE pregunta cuál de los temas desea revisar para la retroalimentación Fig. 5.

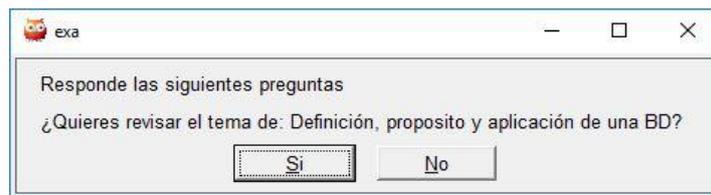


Fig. 5. Ventana que pregunta al alumno por el tema a revisar para la retroalimentación.

Cada uno de los temas tiene una base de hechos formado por las preguntas a realizar como se muestra a continuación:

introducción:-tema_introducción,

pregunta('¿Quieres revisar el tema de: Definición, propósito y aplicación de un BD?'),

pregunta('Un dato puede ser cualquier símbolo, número o letra que no tiene ningún significado.'),

pregunta('Un registro es un conjunto de campos con datos o información'),...

modelo_de_datos:-tema_modelo_de_datos,

pregunta('¿Quieres revisar el tema de: Modelo de datos?'),

pregunta('Los modelos de datos son herramientas que permiten modelar los datos y obtener un modelo de la realidad'),...

bdrelacionales:-tema_bdrelacionales,

pregunta('¿Quieres revisar el tema de: Base de datos relacionales?'),

pregunta('Las tablas son objetos fundamentales de este tipo de BD y están formados por registros y campos.')

Durante la retroalimentación el SE va verificando que el alumno conteste correctamente la mayoría de las preguntas, en caso contrario el SE corta la retroalimentación indicándole que es indefinido el aprendizaje que obtuvo, para esto se ocupa la regla 4 que se muestra como indefinido en la Fig. 4.

5. Conclusiones y trabajo a futuro

La educación en la actualidad exige docentes bien preparados y capacitados, además de que en la mayoría de las instituciones no se asignan de acuerdo a sus perfiles a las materias impartidas, debido a dichas exigencias y limitaciones.

Por una parte con la creación del REA de Fundamentos de bases de datos se obtiene un recurso con información válida y confiable que apoya la educación de los alumnos y la labor de enseñanza de los docentes.

El SE que se propone como se mencionó anteriormente se encuentra en fase de prototipo por lo cual aún no se ha evaluado el impacto, sin embargo se pretende realizar una prueba piloto con un grupo de Educación Media Superior que cursan la materia de Base de Datos y por medio de esta prueba realizar las evaluaciones correspondientes que permitan mejorar el SE en cuanto a su accesibilidad, comprensión, uso y que su contenido sea el ideal con respecto al REA.

La falta de expertos humanos en el ámbito educativo hace factible la necesidad de trabajar con SE que refuercen el aprendizaje de los alumnos proporcionándoles una herramienta que pueden trabajar en la escuela, en la casa o en algún lugar donde tengan acceso a una computadora teniendo la certeza que obtendrán una retroalimentación veraz y confiable, como si estuvieran de manera presencial con el docente.

Actualmente con toda la apertura tecnológica y de recursos educativos en línea se ha suscitado una revolución de enseñanza-aprendizaje virtual, generando infinidad de información que no está revisada ni estandarizada por lo que la elaboración de REA y SE permite ofrecer una educación de calidad con el uso de las herramientas apropiadas.

A largo plazo como dice Matkin [21] las instituciones educativas deben desarrollar y mantener repositorios abiertos de recursos educativos para compartir y reutilizar material educativo, por la misma razón las instituciones también debe generar sus propios REA y fomentar el uso de estos recursos en los docentes, alumnos, y personal que labora en la institución así como en otras instituciones.

Como se puede observar al final el SE se puede considerar un REA, por lo cual se puede centrar el trabajo en el desarrollo de SE como recurso total e individual, con los estándares necesarios para cubrir dicho objetivo.

El SE con los cambios adecuados puede evaluar al alumno, con lo que el docente tendría una evidencia de que realmente se está usando el SE además de que también conocería el nivel de aprovechamiento en cuanto a aprendizaje que ha estado obteniendo el alumno.

Agradecimientos. Los autores agradecen a la Universidad Autónoma del Estado de México y el autor principal agradece al CONACYT por el apoyo otorgado.

Referencias

1. UNESCO. Comunicación e Información. Recursos Educativos Abiertos, <http://www.unesco.org/new/es/communication-and-information/access-to-knowledge/open-educational-resources/>
2. Mortera, G.F.J., Salazar, R.A.L., Rodríguez, G.J., Pérez, N.J.A.: Guía de Referencia para el uso de Recursos Educativos Abiertos [REA] y Objetos de Aprendizaje [OA]. México: CUDI-CONACYT (2011)
3. Celaya, R.R., Lozano, M.F., Ramírez, M.M.S.: Apropiación tecnológica en profesores que incorporan recursos educativos abiertos en educación media superior. RMIE Revista Mexicana de investigación educativa, Vol. 45, pp. 213–487 (2010)
4. TEMOA. Portal de Recursos Educativos Abiertos, <http://www.temoa.info/es>
5. CLARISE. Comunidad Latinoamericana Abierta Regional de Investigación Social y Educativa, <https://sites.google.com/site/redclarise/>
6. RI UAEMEX, <http://ri.uaemex.mx/>
7. Peña, A.A.: Sistemas basados en conocimiento: Una Base para su Concepción y Desarrollo. Instituto Politécnico Nacional México (2006)
8. Castillo, E., Gutiérrez, J.M., Hadi, A.S.: Sistemas Expertos y Modelos Probabilísticos. España (s.f.)
9. Giarratano, J., Riley, G.: Sistemas Expertos: Principios y programación. International Thomson Editores México (2001)
10. Inteligencia Artificial. Extracción de ejemplos de Sistemas Expertos. Universidad Nacional Autónoma de México. <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/219/A7.pdf?sequence=7>
11. Montiel, L., Riveros, V.: Los sistemas expertos en el ámbito educativo. Omnia, pp. 11–28 (2014)
12. Amador, H.L.: Inteligencia artificial y sistemas expertos. Servicio de Publicaciones de la Universidad de Córdoba (1997)
13. Vílchez, Q.E.: Sistemas Expertos para la enseñanza y el aprendizaje de la matemática en la educación superior. Año 2, No. 3, pp. 45–67. Cuadernos de investigación y formación en educación matemática (2007)
14. UNAM-Wolfram Mathematica.: Preguntas frecuentes, <http://www.software.unam.mx/preguntas.html>
15. Cobo, P., Fortuny, J.M.: AgentGeom: un sistema tutorial para el desarrollo de competencias argumentativas de los alumnos a través de la resolución de problemas. Matematicalia revista digital de divulgación matemática, Vol. 3, No. 3 (2007)
16. Ramírez, V.L.A.: Modelo tecnológico para la creación de un recurso educativo abierto (REA). Universidad Autónoma del Estado de México (2014)
17. Pencil Project. An open-source GUI prototyping tool, <http://pencil.evolus.vn/>
18. eXeLearning.net. El Nuevo eXeLearning... <http://exelearning.net/>
19. SWI-Prolog, <http://www.swi-prolog.org/>
20. UAEM Bachillerato Universitario: Programa de la Asignatura: Informática Administrativa, sexto semestre, <http://denms.uaemex.mx/programas/>
21. Matkin, G.W.: The Distance Educator's Opportunity for Institutional Leadership. Continuing Higher Education Review. University of California, Irvine CA, Vol. 74 (2010)

Aplicación de procesos Markovianos para recomendar acciones pedagógicas óptimas en tutores inteligentes

Hermilo Victorio Meza¹, Manuel Mejía-Lavalle¹,
Alicia Martínez Rebollar¹, Obdulia Pichardo², Grigori Sidorov²

¹ Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET),
Tecnológico Nacional de México, Cuernavaca, Morelos,
México

² Instituto Politécnico Nacional (IPN), Ciudad de México,
México

{hvictoriom, mlavalle, amartinez}@cenidet.edu.mx
ayilina@hotmail.com, sidorov@cic.ipn.mx

Resumen. Se fundamenta y describe un proyecto de investigación en desarrollo en el área de Sistemas Tutores Inteligentes que busca automatizar la recomendación de acciones pedagógicas óptimas para que un estudiante lleve a cabo de la mejor manera su aprendizaje. Para lograr lo anterior se están empleando procesos markovianos, específicamente los llamados Procesos de Decisión de Markov Parcialmente Observables, debido a que ellos han permitido generar automáticamente políticas óptimas exitosas en gran variedad de dominios del mundo real. Sin embargo, este tipo de procesos markovianos tienen la problemática de ser de alta complejidad computacional, por lo que, antes de poder ser empleados en problemas reales del área de Tutores Inteligentes, se deberá trabajar en reducir tal complejidad. En el presente artículo se presentan una serie de experimentos que buscan evaluar la complejidad de cuatro métodos o algoritmos exactos y tres aproximados disponibles en la Web para la solución de modelos markovianos. De la evaluación así realizada proponemos varias alternativas para reducir la complejidad de estos algoritmos, pero buscando no afectar las ventajas relevantes que ofrecen los Procesos de Decisión de Markov Parcialmente Observables.

Palabras clave: Sistemas tutores inteligentes, modelo del estudiante, modelo del tutor, procesos de decisión de Markov parcialmente observables.

Application of Markovian Processes for Recommendation of Optimal Educational Actions for Intelligent Tutoring Systems

Abstract. We describe a project still in development about Intelligent Tutoring Systems and optimal educational actions. Good pedagogical actions are key components in all learning-teaching schemes. Automate that is an important

Intelligent Tutoring Systems objective. We propose apply Partially Observable Markov Decision Process (POMDP) in order to obtain automatic and optimal pedagogical recommended actions in benefit of human students, in the context of Intelligent Tutoring System. To achieve that goal, we need previously create an efficient POMDP solver framework with the ability to work with real world tutoring cases. At present time, there are several Web available POMDP open tool solvers, but their capacity is limited, as experiments showed in this paper exhibit. In this work, we describe and discuss several design ideas toward obtain an efficient POMDP solver, useful in our problem domain

Keywords: Intelligent tutoring systems, student model, tutor model, partially observable Markov decision processes.

1. Introducción

Los Sistemas Tutores Inteligentes (STI) son el equivalente digital de los maestros humanos, dado que ofrecen servicios para apoyar al aprendizaje; y, de manera similar a como lo haría un tutor humano experimentado, los STI tienen la capacidad de adaptar las actividades de aprendizaje de acuerdo con la situación de cada estudiante en particular, de tal manera que el proceso se desarrolle en las mejores condiciones que propicien un aprendizaje eficiente. Estos tutores monitorean y entrenan a los estudiantes basados en representaciones tanto de contenido como de aspectos sociales y razonan acerca de la probabilidad de las acciones del estudiante. El campo de la Inteligencia Artificial en la educación investiga sobre la naturaleza del conocimiento y su representación, la forma en que se puede ayudar a aprender a un estudiante y sobre cuáles estilos de interacción de enseñanza son efectivos y cuándo deberían usarse [1].

En el presente artículo se fundamenta y describe un proyecto de investigación, actualmente en desarrollo, en el área de STI que busca automatizar la recomendación de acciones pedagógicas óptimas para que un estudiante lleve a cabo de la mejor manera posible su aprendizaje en línea. Para lograr lo anterior se están empleando los Procesos de Decisión de Markov Parcialmente Observables (POMDP por sus siglas en inglés), debido a que existen múltiples trabajos publicados que reportan que usando estos algoritmos markovianos se han podido generar automáticamente políticas óptimas exitosas en gran variedad de dominios del mundo real como la robótica. Sin embargo, este tipo de procesos markovianos tienen como problema el ser de alta complejidad computacional, es decir, para casos relativamente pequeños el tiempo de procesamiento es alto y conforme el caso crece, el tiempo de respuesta se incrementa mucho más, haciendo a estos procesos prohibitivos para casos del mundo real, donde el número de variables asciende a algunas centenas de variables. Es por eso que, antes de poder ser empleados los POMDP en problemas reales del área de STI, se deberá trabajar en reducir tal complejidad. En el presente artículo se presentan una serie de experimentos que buscan evaluar la complejidad de cuatro métodos o algoritmos (en inglés se les nombra como *solvers*) exactos y tres aproximados que se obtuvieron de la Web y que sirven para la solución de estos modelos markovianos. Con estos experimentos realizados, y que mostramos en el presente artículo, comprobamos las limitaciones que actualmente tienen estas herramientas computacionales. Observando estas limitaciones

nosotros proponemos varias alternativas para reducir la complejidad de estos algoritmos, pero buscando no afectar las ventajas relevantes que ofrecen los POMDP.

El resto de este trabajo está organizado de la siguiente manera: en la Sección 2 fundamentamos esta investigación, describiendo los conceptos relevantes; en la Sección 3 revisamos el trabajo previo que se ha realizado a nivel internacional para el tema poco explorado del uso de POMDP en el área de los STI; en la Sección 4 describimos la forma en que nosotros estamos tratando el problema de la recomendación automatizada de acciones pedagógicas usando POMDP; la Sección 5 detalla los experimentos realizados y los resultados obtenidos; finalmente en la sección 6 se ofrecen las conclusiones y el trabajo a realizar en el futuro inmediato.

2. Contexto teórico

En esta Sección presentamos los fundamentos teóricos en torno a nuestro trabajo de investigación, como lo es una descripción del Modelo del Estudiante sin y con POMDP, la definición formal detrás de los POMDP y la manera en que se obtiene una solución, así como el problema de la complejidad en los POMDP.

2.1. Modelo del estudiante

Las características individuales pueden incluir desde los conceptos y/o habilidades dominadas, estilo de aprendizaje y estado emocional, entre otras. Estas características se concentran en el llamado Modelo del Estudiante, componente esencial de los STI, dado que la adaptabilidad del sistema depende de la información contenida en este modelo.

Dado que la característica distintiva de los STI es su adaptabilidad a la situación personal de cada estudiante, la representación del conocimiento se considera un elemento de especial importancia. En particular, el Modelo del Estudiante es el componente del STI que contiene la información necesaria para una adecuada adaptabilidad reflejada en actividades de aprendizaje aplicables a la situación actual. El Modelo del Tutor recibe información de los Modelos del Estudiante y de Dominio y toma decisiones acerca de las estrategias tutoriales y acciones pedagógicas. Basado en el conocimiento indicado debe tomar decisiones tales como si intervenir o no, y de ser así, cuándo y cómo. La planificación y distribución de contenidos son también parte de las funciones del Modelo del Tutor [2].

El Modelo del Estudiante es el componente clave de un STI. Idealmente, debe contener tanto conocimiento como sea posible acerca de los estados cognitivo y afectivo del estudiante y su evaluación conforme avanza el proceso de aprendizaje. El modelo del estudiante incorpora varias funciones: a) debe reunir información sobre el estudiante; b) debe crear una representación del conocimiento del estudiante y de su proceso de aprendizaje a partir de esa información; y c) debe tomar en cuenta esos datos para realizar algún tipo de diagnóstico, tanto del estado de conocimiento del estudiante y en términos de seleccionar estrategias pedagógicas óptimas para presentar subsecuente información de dominio para el estudiante [2].

La información contenida en el Modelo del Estudiante determina el comportamiento del sistema, generando una estrategia de adaptación basada en las características personales del estudiante que atiende, en un momento determinado.

De los diversos métodos que permiten modelar las características del estudiante se seleccionó a los Procesos de Decisión de Markov Parcialmente Observables (Partially Observable Markov Decision Processes o POMDP). Se considera que no se tiene acceso directo al conocimiento del estudiante (procesos y planes mentales) [3], sino que se deduce a partir de ciertas evidencias como lo son las evaluaciones que se le apliquen, por lo que se tiene cierto grado de incertidumbre. Entre los métodos de modelado que consideran el manejo de la incertidumbre se cuentan a la Lógica Difusa, Redes Bayesianas y similares como las Redes Bayesianas Dinámicas, Redes de Decisión, etc. y los modelos de Markov, como las Cadenas de Markov, los Modelos Ocultos de Markov, los Procesos de Decisión de Markov (completamente observables) y los Procesos de Decisión de Markov Parcialmente Observables ó POMDP.

2.2. Modelado del estudiante con POMDP

Los métodos de modelado citados, con excepción de los dos tipos de procesos de decisión de Markov, requieren de componentes adicionales que determinen las acciones que deba realizar el sistema una vez determinado el estado de conocimiento del estudiante. Los Procesos de Decisión de Markov (completa y parcialmente observables) incluyen en su estructura las acciones que puede ejecutar el sistema, siendo guiadas por las recompensas indicadas en el modelo. Al no ser completamente observable el estado de conocimiento del estudiante se descartan los Procesos de Decisión de Markov Completamente Observables (MDPs o COMDPs).

El proceso de enseñanza puede modelarse como un problema de toma de decisiones secuenciales bajo incertidumbre. Un STI debe seleccionar las acciones pedagógicas, tales como acciones remediales y recomendaciones, sin tener un conocimiento absoluto de los estados mentales del estudiante y sin garantía de que las acciones tendrán el efecto deseado [3].

Al enfrentarse con un ambiente parcialmente observable, se justifica el uso de un modelo general para problemas de decisión secuencial como los POMDP. Un POMDP es un modelo para planeación bajo incertidumbre, lo que da al agente la capacidad de estimar efectivamente el resultado de sus acciones aún cuando no puede observar el ambiente con exactitud.

2.3. Definición formal de POMDP

Un POMDP es un formalismo para representar problemas que involucran la planeación bajo incertidumbre. Se puede usar para computar iterativamente una política para optimizar una función de recompensas sobre horizontes de tiempo progresivamente largos. Han probado ser útiles en muchas tareas de planeación diferentes a los STI tales como el control de robots, planeación de tratamientos médicos e interpretación de voz (*spoken dialogue*) o video. El propósito del POMDP es describir cómo cambian los estados mediante la ejecución de acciones, de tal manera que en el proceso de su solución se pueda estimar el estado y cuál acción se aplica en el momento y se propongan planes para mejorar el estado a lo largo del tiempo [4].

La composición de un POMDP se define como una tupla $(S, A, P(s'|s,a), R(a,s), O, P(o/s,a))$ que comprende: un conjunto de estados ocultos S ; un conjunto de acciones A ; un conjunto de probabilidades de transición $P(s'|s,a)$ que definen cómo las acciones cambian los estados del sistema; un conjunto de recompensas $R(a,s)$ que asignan valores para los estados alcanzados o acciones completadas; un conjunto de observaciones O y un conjunto de probabilidades de emisión $P(o/s,a)$ que definen cuáles observaciones aparecen cuando el sistema se encuentra en un estado dado. La Figura 1 ilustra estas ideas.

2.4. Solución de un POMDP

La solución de un POMDP consiste en determinar la política o conjunto de acciones a ejecutar que producirá la mayor recompensa, aplicables a lo que se perciben como los diferentes estados del sistema. Como consecuencia de una acción aplicada, el sistema hará probablemente una transición a otro estado.

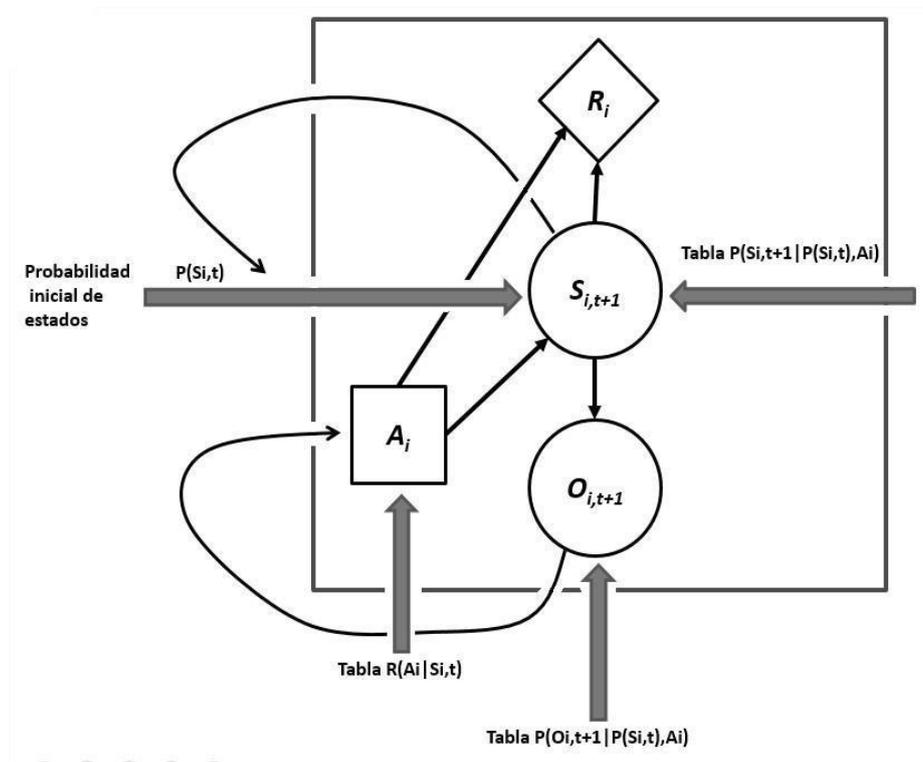


Fig. 1. Representación gráfica de un POMDP.

Para resolver los POMDP hay una amplia diversidad de algoritmos que, al mismo tiempo que pretenden obtener una solución de calidad aceptable (que no siempre es la óptima) buscan aliviar el problema de complejidad en espacio de almacenamiento y tiempo de procesamiento que tiene este método de representación del conocimiento [5, 6, 7].

2.5. Problemática de los POMDP

Muchos problemas pueden modelarse con POMDP, pero muy pocos pueden resolverse hasta alcanzar el óptimo debido a su complejidad computacional. La Teoría de la Computación clasifica los problemas de acuerdo con su dificultad computacional y denomina de clase P a los problemas fáciles o tratables que pueden resolverse en una cantidad de tiempo polinomial; en la clase NP , más amplia, y que incluye a P , se encuentran los problemas que se consideran difíciles o intratables; los problemas NP -**completos** son los problemas más difíciles de la clase NP y se asume que tardan una cantidad exponencial de tiempo. La cualidad de completitud se refiere al conjunto de problemas más difícil de resolver en la clase. $PSPACE$ es otra clase aún más amplia que se refiere a los problemas que pueden resolverse en una cantidad de espacio polinomial. La clase P se incluye en $PSPACE$ dado que cualquier algoritmo que corre en un tiempo polinomial sólo puede consumir una cantidad polinomial de recursos [5]. Aún cuando los problemas modelados como Procesos de Decisión de Markov Completamente Observables (COMDPs) se clasifican como problemas P -**completos**[19], no es el caso de los POMDP, que se clasifican como $PSPACE$ -**completos** [8] aún para horizontes finitos muy reducidos [19] por lo que sólo queda el recurso de buscar soluciones aproximadas [5]. Algunos problemas con solución exacta difícil tienen un correspondiente algoritmo de aproximación fácil y hay otros cuya aproximación es tan difícil como la solución exacta, siendo desafortunadamente éste el caso para los POMDP, aún para horizontes finitos [5].

El principal problema con los POMDP es que su complejidad los hace aplicables, aparentemente, sólo en ambientes pequeños [7, 9]. Para los POMDP sólo pueden encontrarse soluciones óptimas (estrategias o políticas que indican acciones a ejecutar según el estado actual) para problemas con unos pocos estados (menos de 20, contra 100 o más necesarios para un STI, según [3]) por sus requerimientos de almacenamiento y tiempo de procesamiento. Por lo tanto, la complejidad de los procedimientos de solución del problema se convierte en el aspecto clave de la aplicación exitosa del modelo en problemas del mundo real, aún a expensas de la optimalidad. Sin embargo, la mayoría de los problemas de interés tienen un espacio de estados enorme, lo que motiva a la búsqueda de métodos aproximados [8, 9].

Debido a que los POMDP consideran el manejo de la incertidumbre en la representación del conocimiento y deciden, a largo plazo, las acciones a ejecutar en base a las recompensas, parece el método de modelado natural para los STI y sin embargo tienen una muy escasa aplicación en éstos, la cual se ha dado sólo en tiempos recientes [3, 10, 12, 13, 14].

3. Trabajo relacionado con la aplicación de los POMDP en educación

Aunque los POMDP ofrecen ventajas frente a otros métodos de modelado del estudiante, en los STI han sido poco investigados en esta área y eso sólo recientemente [3, 12, 10, 14]. Las investigaciones relacionadas con POMDP generalmente experimentan con solución de problemas relacionados con la navegación robótica [8, 9, 7, 5], aunque se menciona que también pueden aplicarse en problemas relacionados

con la operación, mantenimiento y reparación de maquinaria industrial y en medicina en el tratamiento de las enfermedades [15]. La causa principal de esta escasa aplicación se deriva del gran problema de los POMDP: la complejidad de éstos al tener grandes requerimientos de almacenamiento y tiempo de procesamiento. La solución óptima para un POMDP sólo puede determinarse para problemas pequeños de 10 a 20 estados como máximo, lo cual dejaría fuera a los problemas del mundo real con una cantidad mucho mayor de estados, como sería el caso de los STI. Existen métodos que encuentran soluciones aproximadas, que se espera sean aceptables y cercanas a la óptima, pero aún así siguen siendo complejas y, posiblemente, no aplicables a todo tipo de problemas [9, 7, 8, 10].

Las investigaciones relacionadas con POMDP en STI han considerado la complejidad del modelo y han aplicado métodos que hagan tratable la solución. Aunque se llega a mencionar que los métodos utilizados pueden tratar con POMDP de tamaño significativo, en su investigación han abordado sólo modelos de tamaño relativamente pequeño.

Por ejemplo, Folsom-Kovarik [3, 4] combina un simulador de adiestramiento militar con un STI basado en POMDP. El STI tiene el objetivo de resolver problemas de aprendizaje (*knowledge gaps*) derivados de la operación del simulador. Para reducir la complejidad del POMDP se plantean dos estrategias: colas de prioridad y cadenas de observación.

Theocharous [12] desarrolló un STI para enseñar a ordenar un conjunto de monedas virtuales. Considera que la enseñanza es un dominio interesante para la Inteligencia Artificial porque deben resolverse tres problemas al mismo tiempo: el maestro debe inferir variables sociales (atención, aburrimiento, confusión, conocimientos, aptitudes) y variables físicas (progreso de la tarea, objetos usados, actividad actual) y debe combinar este conocimiento para planear estrategias de interacción efectivas momento-a-momento. El algoritmo cambia entre estados prototípicos de estudiantes basándose en la probabilidad de las variables sociales. Es decir, se crean varios POMDP pequeños que pueden ser seleccionados por un segundo POMDP, un POMDP jerárquico.

Rafferty [10, 11] considera tres modelos de aprendizaje del estudiante (modelo sin memoria, modelo discreto con memoria, modelo continuo) y se presentan métodos aproximados para encontrar las acciones de enseñanza óptimas dados los grandes espacios de estados y acciones que se presentan en la enseñanza. Para determinar la política (que seleccione la mejor acción dada una distribución sobre el estado de conocimiento actual del estudiante, el *beliefstate*) se seleccionó una técnica *online* de búsqueda hacia adelante (*forward*), dado que este tipo de técnicas ha demostrado tener un buen desempeño en otros grandes dominios.

Zhang [13] y Wang [14] presentan un marco de POMDP para construir un STI en el que se puede hacer una reducción del espacio de estados posiblemente exponencial a un espacio manejable; la reducción presentada se basa en que en un sistema de aprendizaje de conceptos hay un orden de prerrequisitos, de tal manera que el dominio de un concepto cualquiera generalmente depende de conceptos previos; para llegar al estado de dominio del concepto considerado, se debió pasar, previamente, por los estados anteriores a él (prerrequisitos) con lo que se evitarían estados (combinaciones) imposibles.

4. Dominio de la investigación: Automatizar la recomendación de acciones pedagógicas óptimas

Para realizar la experimentación se diseñaron modelos aleatorios de POMDP de tamaño $2 \times 2 \times 2$ (es decir: 2 estados, 2 acciones y 2 observaciones), $3 \times 3 \times 3$, $4 \times 4 \times 4$, $5 \times 5 \times 5$, $6 \times 6 \times 6$, $10 \times 10 \times 10$ y $20 \times 20 \times 20$. Estos modelos los llamamos aleatorios porque generamos previamente y con un programa de computadora, valores aleatorios que usamos como las probabilidades de transición de un estado al siguiente.

También se diseñaron modelos de POMDP que consideran 50% de eficiencia en el aprendizaje, de tamaño $2 \times 2 \times 2$, $3 \times 3 \times 3$, $4 \times 4 \times 4$, $5 \times 5 \times 5$ y $6 \times 6 \times 6$.

Además se diseñó un modelo de POMDP de $10 \times 10 \times 10$ que considera la eliminación de probabilidades de transición y observaciones para probabilidades menores a 0.05 (umbral de 5%), es decir, se traducen a 0.0. Después de la modificación de estas probabilidades se normalizan el resto de los valores de la fila, para que sumen 1.0.

Los modelos propuestos en las pruebas actuales sólo consideran una acción por estado. En una situación ideal se supone que la acción $a1$ aplicada en el estado $s1$ genera la transición al estado $s2$.

Se considera una distribución uniforme de recompensas (todos los estados tienen el mismo valor o proporción). De lo anterior, la recompensa (calculada como porcentaje) de aplicar la acción i en el estado i de un modelo de n estados es $r = 100 / (n - 1)$. Se divide entre $(n - 1)$ porque se asume que el último estado es el objetivo y no hay recompensas por aplicar acciones adicionales en este estado. Cualquier acción x diferente de i aplicada en el estado i no generará recompensa (0). Aunque los modelos podrían considerar castigos o costos (recompensas negativas) por aplicar acciones que generen transiciones a estados no deseables.

Con la presente investigación se espera encontrar un procedimiento de solución de POMDP que supere a los hasta ahora usados para STI, sin afectar de manera importante la representatividad del modelo.

5. Resultados de la experimentación

Al momento de redactar el presente artículo se han hecho pruebas iniciales que consideran algunos de los algoritmos de solución aplicados a modelos de POMDP de tamaño relativamente reducido. Se utilizó el software *Pomdp-solve v5.4* [16] para calcular las soluciones exactas (*Enumeration, Twopass, Witness, Incremental Pruning*) de los modelos generados. También se utilizó el paquete de software *ZMDP v1.1.7* [17] que ejecuta los algoritmos aproximados *HSVI* y *FRTDP*. Además, se utilizó el paquete *APPL v0.96* [18] que aplica el algoritmo aproximado *SARSOP*. Todas las sesiones de experimentación se dejaron en ejecución por un máximo aproximado de una hora (3,600 segundos). El software se ejecutó bajo el sistema *Linux Ubuntu 15*. Se utilizó un equipo de cómputo portátil con procesador *Intel CORE i5* a 2.40GHz y 4GB en RAM. En la Tabla 1 se muestran los resultados obtenidos aplicando métodos de solución exacta.

Tabla 1. Solución de modelos de POMDP mediante métodos exactos.

MODELO	MÉTODOS EXACTOS											
	ENUMERATION			TWO PASS			WITNESS			INCREMENTAL PRUNING		
	t(seg)	obs.**	Lim. Inf.	t(seg)	obs.**	Lim. Inf.	t(seg)	obs.**	Lim. Inf.	t(seg)	obs.**	Lim. Inf.
2aleat	0.098		3481.403000	<u>0.092</u>		3481.403000	0.100		3481.403000	0.113		3481.403000
2ef	0.305		1753.302000	0.090		1753.302000	<u>0.072</u>		1753.302000	0.105		1753.302000
3aleat	0.319		1108.095000	0.138		1108.095000	0.130		1108.095000	<u>0.108</u>		1108.095000
3ef	4.430		1282.275000	21.731		1282.275000	3.918		1282.275000	<u>2.903</u>		1282.275000
4aleat	2.634		712.418000	7.787		712.418000	2.832		712.418000	<u>2.478</u>		712.418000
4ef	404.950		168.117000	428.390		168.117000	53.040		168.117000	<u>23.960</u>		168.117000
5aleat	*	3 epocas	42.257000	*	4 epocas	49.636000	*	4 epocas	49.636000	*	<u>15 epocas</u>	122.051000
5ef	*	3 epocas	50.659000	*	3 epocas	50.659000	*	4 epocas	60.291000	*	<u>5 epocas</u>	69.499000
6aleat	*	2 epocas	27.691942	*	2 epocas	27.691942	*	<u>3 epocas</u>	33.449471	*	<u>3 epocas</u>	33.449471
6ef	*	2 epocas	31.700000	*	2 epocas	31.700000	*	<u>3 epocas</u>	39.205633	*	<u>3 epocas</u>	39.205633
10aleat	*	<u>2 epocas</u>	2.650088	*	<u>2 epocas</u>	2.650088	*	<u>2 epocas</u>	2.650088	*	<u>2 epocas</u>	2.650088
20aleat	*	<u>1 epoca</u>	0.525263	*	<u>1 epoca</u>	0.525263	*	<u>1 epoca</u>	0.525263	*	<u>1 epoca</u>	0.525263

* No se obtuvo solución completa en 3600 segundos (1 hora)

** Iteraciones resueltas en tiempo límite de 3600 segundos (1 hora)

Nótese que sólo se obtuvieron soluciones exactas completas para los modelos de tamaño 2 a 4. Para los restantes modelos se obtuvieron soluciones parciales. Se observan los mejores tiempos de solución (subrayados en la Tabla 1) para el método *Incremental Pruning*. Los peores tiempos se obtuvieron con el método *Two Pass*. La Figura 2 ilustra los resultados obtenidos para los modelos de tamaño 2 a 4.

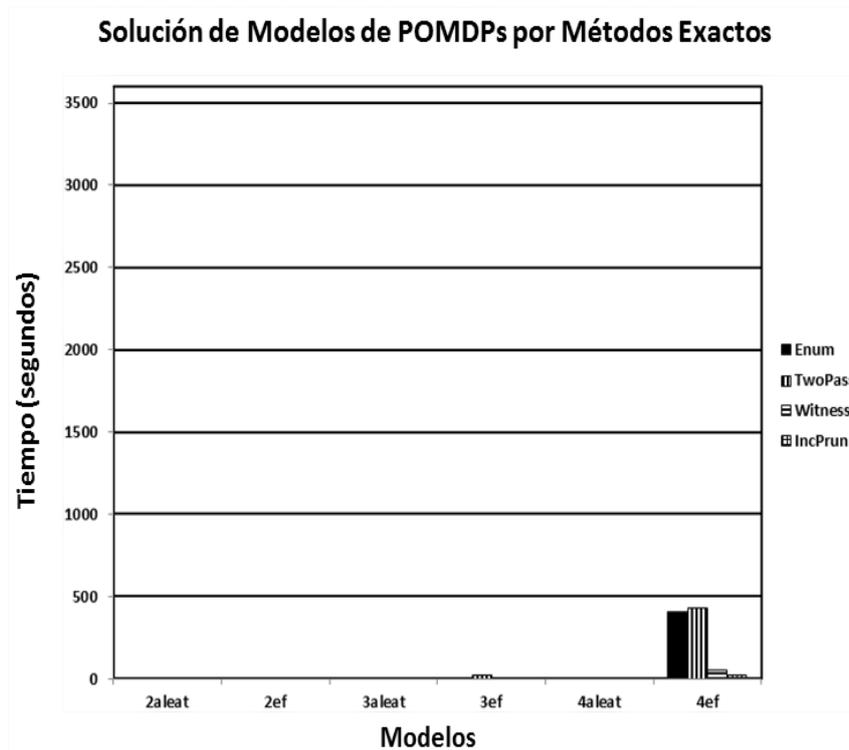


Fig. 2. Resultados con métodos exactos (escala lineal).

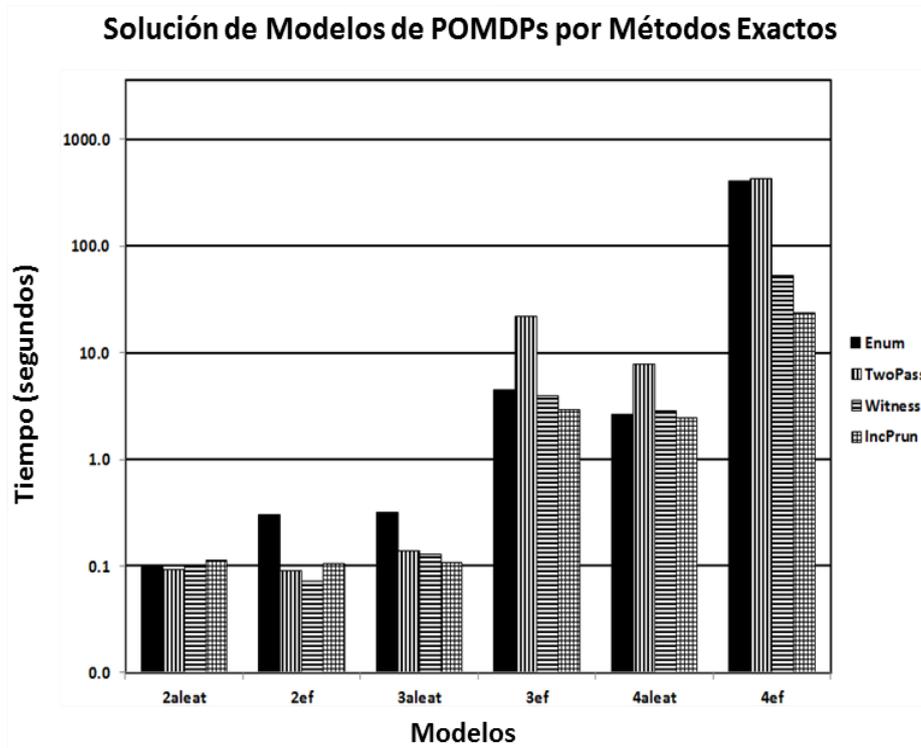


Fig. 3. Resultados con métodos exactos (escala logarítmica)

La Figura 3 muestra los mismos resultados pero con una escala logarítmica en el eje y, para observar mejor los resultados.

La Tabla 2 muestra los resultados obtenidos aplicando métodos de solución aproximados.

Tabla2. Solución de modelos de POMDP mediante métodos aproximados.

MODELO	MÉTODOS APROXIMADOS											
	SARSOP				FRTDP				HSVI			
	t(seg)	Lim. Inf.	Lim. Sup.	Precision	t(seg)	Lim. Inf.	Lim. Sup.	Precision	t(seg)	Lim. Inf.	Lim. Sup.	Precision
2aleat	0.009	3484.860000	3484.860000	0.000476	0.000	3484.857700	3484.858100	0.000476	0.000	3484.857700	3484.858100	0.000439
2ef	0.010	1754.970000	1754.970000	0.000475	0.000	1754.966400	1754.966900	0.000475	0.000	1754.966400	1754.966900	0.000438
3aleat	212.190	1109.170000	1109.170000	0.000990	2196.000	1109.170700	1109.171600	0.000865	1221.000	1109.170700	1109.171700	0.001000
3ef	3600.000	1283.530000	1283.610000	0.083941	*** error desconocido ***				3600.000	1283.528700	1283.612900	0.084127
4aleat	3600.000	713.105000	714.704000	1.599830	3600.000	713.104500	715.840600	2.736080	3600.000	713.104500	714.297700	1.193130
4ef	3600.000	168.118000	168.147000	0.029339	3600.000	168.117700	168.301800	0.184090	3600.000	168.117700	168.147100	0.029417
5aleat	3600.000	412.872000	420.809000	7.936890	3600.000	412.871800	421.218700	8.346880	3600.000	412.871800	416.356700	3.484870
5ef	3600.000	507.242000	534.347000	27.105000	3600.000	507.242100	535.308100	28.066000	3600.000	507.242200	526.118500	18.876400
6aleat	3600.000	313.586000	346.178000	32.591900	3600.000	313.585700	349.568400	35.982700	3600.000	313.586000	337.602700	24.016700
6ef	3600.000	95.507200	96.874200	1.366960	3600.000	95.507200	98.076600	2.569340	3600.000	95.507200	97.257400	1.750190
10aleat	3600.000	3.113280	3.264020	0.150742	3600.000	3.112900	3.486400	0.373573	3600.000	3.113300	3.294400	0.181095
20aleat	3600.000	2.153080	2.800530	0.647453	*** error desconocido ***				3600.000	2.163100	2.703000	0.539856

Se observa que el método *SARSOP* obtiene los mejores tiempos de solución para el modelo *3aleat*. Los mejores valores (menores) de precisión corresponden al método *HSVI*. En el método *FRTDP*, además de presentar menos elementos favorables

(subrayados en la Tabla 2) tiene problemas con algunos modelos (*3ef* y *20 aleat*). La Figura 4 muestra los resultados obtenidos para los modelos de tamaño 2 a 4, comparables con la Figura 2.

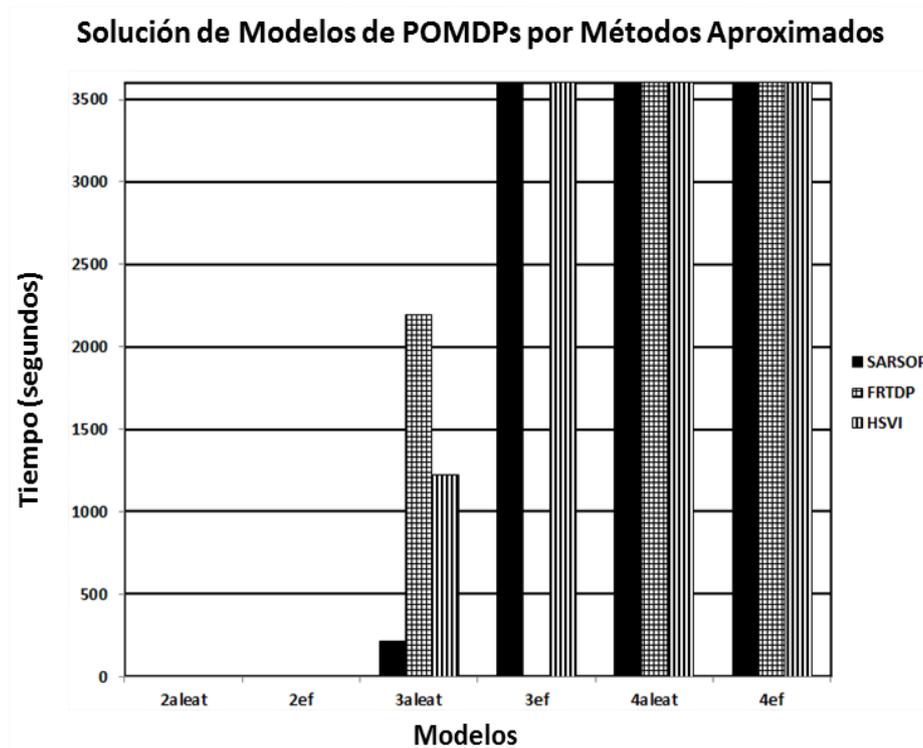


Fig. 4. Resultados con métodos aproximados.

Hacemos notar que, hasta donde podemos analizar, los métodos exactos comparan las diferencias en recompensa entre la última época (iteración) calculada y la penúltima. Si esta diferencia es menor a la precisión buscada se detiene el proceso y se genera la política resultante. Por otro lado, en los métodos aproximados se comparan las diferencias en recompensa entre la última época (iteración) calculada (a la que se llama *lower bound* o límite inferior) y un límite superior (*upper bound*) que es un máximo teórico calculado, que siempre es superior a la última época calculada. Si esta diferencia es menor a la precisión buscada se detiene el proceso y se genera la política resultante. El tiempo extra que utilizan los métodos aproximados es para la búsqueda de una mayor recompensa tendiendo hacia el límite superior cuando no se ha alcanzado la precisión establecida.

Además, la comparativa sólo se aplica a los modelos de tamaño 2 a 4, pues para modelos de tamaño 5 o mayor los métodos exactos no completan la solución y sólo resuelven unas pocas épocas. En cambio, los métodos aproximados sí generan una solución, aunque la precisión lograda varía dependiendo del tamaño del modelo y del tiempo de procesamiento.

Algo similar podría observarse con el método *Incremental Pruning* al compararlo con los demás métodos exactos: en general es el más rápido, pero lo superan todos los demás en los modelos de tamaño 2 (*Zaleat*, 2ef).

Como consecuencia de observar los resultados de la experimentación, proponemos algunas ideas de diseño tendientes a reducir la complejidad en el proceso de solución del modelo de POMDP en tres aspectos:

- Descartar transiciones de estados (o combinaciones de transición*recompensa, transición*recompensa*observación) menores a umbrales mínimos establecidos.
- Clasificar estados en un máximo de tres grandes grupos: retroceso (recompensas menores a las obtenidas en el estado actual), detenido (recompensas iguales a las obtenidas en el estado actual) y avance (recompensas mayores a las obtenidas en el estado actual) y variaciones en niveles de granularidad (subniveles de los tres grandes grupos, para ampliar abanico de búsqueda).
- Soluciones como POMDP jerárquicos de diferentes niveles comparados contra modelos monolíticos.

6. Conclusiones y trabajo futuro

Se ha presentado y fundamentado una investigación que se encuentra actualmente en desarrollo y que busca automatizar la recomendación de acciones pedagógicas óptimas en el contexto de los STI y por medio del uso de POMDP.

Dada la complejidad computacional de los POMDP hemos realizado una serie de experimentos con algoritmos (*solvers*) que proporcionan soluciones exactas y aproximadas y que fueron obtenidos libremente de la Web. En la experimentación realizada estos *solvers* mostraron sus limitaciones, según reportamos en la Sección 5.

Por ejemplo, al emplear métodos exactos se obtuvieron soluciones para los modelos de tamaño de 4 estados e inferiores. Para los modelos de tamaño mayor se obtuvieron soluciones parciales (épocas resueltas) y en algunos casos se agotó la memoria disponible del equipo de cómputo.

Al emplear métodos aproximados se obtuvieron soluciones con precisión aceptable para los algoritmos *HSVI*, *FRTDP* y *SARSOP*, obteniéndose las mejores soluciones para este último. El algoritmo *FiniteGrid* sólo obtuvo soluciones para modelos de tamaño 6 e inferiores. Al observar estas limitaciones, varias ideas útiles de diseño surgen para reducir la complejidad, sin afectar la calidad de la solución.

En el futuro se deberán probar estas ideas de diseño incorporándolas en los *solvers* libremente disponibles en la Web para validarlos, a saber: a) Descartar transiciones de estados, b) Clasificar estados en un máximo de tres grandes grupos: retroceso, detenido y avance, más variaciones en subniveles de granularidad, y c) Soluciones como POMDP jerárquicos de diferentes niveles.

Referencias

1. Woolf, B.: Building Intelligent Interactive Tutors. Student-centered strategies for revolutionizing e-learning, Morgan Kaufmann, Burlington (2009)

2. Nkambou, R.: Introduction: What Are Intelligent Tutoring Systems, and Why This Book? In: R. Nkambou et al. (eds.) *Advances in Intelligent Tutoring Systems*, SCI 308, pp. 1–12 (2010)
3. Folsom-Kovarik, J.T., Sukthankar, G., Schatz, S.: Tractable POMDP representations for intelligent tutoring systems. *ACM Trans. Intell. Syst. Technol.* Vol. 4, No. 2, Article 29 (2013)
4. Folsom-Kovarik, J., Sukthankar, G., Schatz, S.: Integrating Learner Help Requests Using a POMDP in an Adaptive Training System. In: *Proceedings of the Twenty-Fourth Innovative Applications of Artificial Intelligence Conference* (2012)
5. Cassandra, A.: *Exact and Approximate Algorithms for Partially Observable Markov Decision Processes*. Ph. D. thesis. Brown University. Providence, Rhode Island (1998)
6. The POMDP Page, <http://www.pomdp.org/> (consultado en abril 2016)
7. Hauskrecht, M.: Value-Function Approximations for Partially Observable Markov Decision Processes. *Journal of Artificial Intelligence Research*, Vol. 13, pp. 33–94 (2000)
8. Paquet, S., Chaib-draa, B., Ross, S.: Hybrid POMDP Algorithms. In *Proceedings of The Workshop on Multi-Agent Sequential Decision Making in Uncertain Domains (MSDM)* (2006)
9. Ross, S., Pineau, J., Paquet, S.: Online Planning Algorithms for POMDPs. *Journal of Artificial Intelligence Research*, Vol. 32, pp. 663–704 (2008)
10. Rafferty, A., Brunskill, E., Griffiths, T., Shafto, P.: Faster Teaching via POMDP Planning. *Cognitive Science*, pp. 1–43 (2015)
11. Rafferty, A.: *Applying Probabilistic Models for Knowledge Diagnosis and Educational Game Design*. Technical Report No. UCB/EECS-2014-61, Electrical Engineering and Computer Sciences, University of California at Berkeley (2014)
12. Theocharous, G., Beckwith, R., Butko, N., Philipose, M.: Tractable POMDP Planning Algorithms for Optimal Teaching in SPAIS. In: *Workshop on Plan Activity, and Intent Recognition (PAIR), IJCAI* (2009)
13. Zhang, P.: *Using POMDP-based Reinforcement Learning for Online Optimization of Teaching Strategies in an Intelligent Tutoring System*. Master of Science thesis. University of Guelph, Guelph, Ontario, Canada (2013)
14. Wang, F.: *POMDP Framework for Building an Intelligent Tutoring System*. School of Computer Science, University of Guelph, Guelph, Canada (2014)
15. Sondik, E., Smallwood, R.: The Optimal Control of Partially Observable Markov Processes Over a Finite Horizon. *Operations Research*, Vol. 21, No. 5, pp. 1071–1088 (1973)
16. pomdp-solve, <http://www.pomdp.org/code/> (consultado en abril 2016)
17. ZMDP software for POMDP and MDP planning, <https://github.com/trey0/zmdp> (consultado en abril 2016)
18. Approximate POMDP Planning Software, <http://bigbird.comp.nus.edu.sg/pmwiki/farm/appl/> (consultado en abril 2016)
19. Papadimitriou, C., Tsitsiklis, J.: The complexity of Markov Decision Processes. *Mathematics of Operations Research*, Vol 12, No. 3, pp. 441–450 (1987)

Propuesta para reducir espacio de búsqueda en entornos virtuales

Víctor Tomás Tomás Mariano, Felipe de Jesús Núñez Cárdenas,
Efraín Andrade Hernández

Universidad Autónoma del Estado de Hidalgo: Escuela Superior de Huejutla,
Huejutla de Reyes, Hidalgo, México

{victor_tomasm, felipe.huejutla, andrade_a_h}@hotmail.com

Resumen. Se propone un proceso de etiquetado de un área de búsqueda tipo rejilla, con la cual se identifican celdas en las que se puede navegar de manera segura. Con las etiquetas se identifican las celdas que están libres de colisiones además permite reducir el número de celdas a recorrer durante el proceso de encontrar una trayectoria o ruta solución que conecta a dos celdas distintas dentro del entorno virtual. Los entornos virtuales generados son de distintos tamaños con salida en formato binario lo que facilita hacer una representación gráfica en 2D.

Palabras clave: Navegación, entorno virtual, espacio de búsqueda.

Proposal for Reduction of the Search Space in Virtual Environments

Abstract. A process of labeling a search of square type grid is proposed. With labels squares that are free of collisions are identified also reduces the number of squares to travel during the process of finding a solution path or route connects starting point to the goal within the virtual environment. The generated environments are different sizes with output in binary format which facilitates make a graphical representation 2D.

Keywords: Navigation, virtual environment, search space.

1. Introducción

El problema de búsqueda es un tema en la que en los últimos años ha tenido bastante auge, principalmente en el área de robótica móvil, en este tipo de problemas es necesario definir en primera instancia el tipo de búsqueda que se pretende realizar, básicamente hay dos tipos: búsqueda informada y búsqueda no informada.

La búsqueda no informada o a ciegas, realiza un recorrido del espacio de búsqueda de una forma sistemática, pero sin tener en cuenta ningún tipo de información sobre el dominio del problema que se está resolviendo [5]. Por lo general utiliza un árbol de búsqueda para representar los distintos estados en los que se puede navegar. En ocasiones este árbol es muy grande que es difícil representarlo y el costo computacional es bastante alto.

Por otro lado, la búsqueda informada admite tener un mecanismo que permite dirigir la búsqueda hacia zonas más prometedoras, de modo que se puede encontrar una solución sin necesidad de visitar tantos nodos de un árbol como los que en general requiere una estrategia de búsqueda a ciegas. Este tipo de búsqueda utilizan heurísticas que son criterios, reglas o métodos que ayudan a decidir cuál es la mejor alternativa entre varias posibles para alcanzar un determinado objetivo [5].

Una parte fundamental para decidir el tipo de búsqueda a realizar, depende del entorno en el que se va a modelar el problema, es decir, el espacio de búsqueda.

Entonces, se puede hablar de espacios de búsqueda continuos y discretos. En este trabajo se modela un espacio de búsqueda discreto que consiste en generar un área rectangular de dimensiones de m por n en la que hay celdas libres y celdas ocupadas (obstáculos). Así mismo, se propone el uso de un conjunto de máscaras para etiquetar el espacio de búsqueda de manera a priori con el objetivo de reducir el número de celdas a explorar en el entorno durante el recorrido para encontrar una solución, otorgando la posibilidad de que el espacio etiquetado esté libre de colisiones.

El artículo está organizado de la siguiente manera, en la sección dos abarca el problema que implica hacer búsquedas y planificación de trayectorias; en la sección tres de los entornos tipo rejilla; en la sección cuatro del proceso de etiquetado propuesto; en la sección cinco de los resultados obtenidos, conclusiones y del trabajo futuro.

2. El problema de hacer búsquedas

El problema de realizar búsquedas se ha analizado en varios trabajos de investigación [3, 4, 7, 8], principalmente en robótica móvil, y la idea general consiste en que un robot móvil debe ser capaz de encontrar una trayectoria o camino que lo conduzca de un punto de partida a otro punto objetivo, dentro de un entorno virtual o real, este robot es capaz de censar en todo momento el entorno para conocer su posición actual, en base a las mediciones censadas debe ser capaz de tomar decisiones para moverse en la dirección que lo acerque más al objetivo perseguido.

En el proceso de encontrar la ruta, el robot debe evitar obstáculos, y de preferencia se desea que el robot se aleje de esos obstáculos de tal manera que la trayectoria que genera esté libre de colisiones [4, 7].

La búsqueda puede llevarse a cabo en dos tipos de entornos: entornos conocidos y entornos desconocidos. En la búsqueda en entornos conocidos, se tiene información a priori de la posición actual u origen, y también de la posición en la que se encuentra el punto final u objetivo perseguido, esto da la posibilidad de tener orientación hacia dónde debe moverse el robot dentro del entorno [7, 8]. Por lo regular, el entorno a explorar permanece fijo. Lo que se pretende es encontrar la ruta o camino que conduzca al punto final.

Por otro lado, la búsqueda en entornos desconocidos, consiste en que el robot solo conoce la posición actual dentro del entorno, el objetivo perseguido también es alcanzar un punto meta, sin embargo, en estos entornos, el robot desconoce en donde está ubicado tal punto, además el entorno es dinámico o cambiante.

Ambos tipos de búsqueda se pueden modelar en espacios de búsqueda discretos o continuos, según sea el problema a resolver, en el caso discreto se tiene un entorno en la que el área de búsqueda es una rejilla rectangular (*pathfinder grids*) con dimensiones definidas en términos de filas y columnas de n por m respectivamente [1].

2.1. Planificación de trayectorias

La planificación se define como la búsqueda de una ruta libre de obstáculos desde una posición inicial hasta otra posición final a través del entorno de trabajo del robot móvil. Esta ruta se realiza mediante el uso de la información que se posee del entorno actualmente. El caso más simple consiste en considerar un entorno conocido en su totalidad y estático. Con estas apreciaciones resulta factible la aplicación de un algoritmo de búsqueda en grafos, que emplee cierta función de coste para la obtención de la ruta. De esta forma, la finalidad de la planificación consiste en localizar una ruta libre de obstáculos, definida como una secuencia de ordenada de objetivos, que conduzca al robot desde la posición inicial hasta la final [4,5].

Modelar el entorno de búsqueda como un grafo, conlleva la necesidad de asociar cada nodo o vértice del grafo a una posición $p[i, j]$ dentro del entorno [9]. La arista que va de v_i a v_j que conecta al vértice i con el vértice j , ambos nodos son adyacentes, es decir, están conectados, existe un camino que va desde el nodo I a otro nodo J , por lo que se dice que es alcanzable y que hay un camino entre ambos nodos.

Esta forma de representar el espacio de búsqueda, consiste en dos etapas, la primera, es construir el grafo asociado al entorno, según sea el caso, el grafo puede ser dirigido, no dirigido o mixto, depende de las restricciones de movilidad en la exploración, este grafo asociado debe ser lo más real o parecido al entorno. En la segunda etapa, se designa un vértice inicial como punto de partida y vértice final como punto que se desea alcanzar, en esta etapa se aplican algoritmos para recorrido de grafos que dan buenos resultados para encontrar el camino entre dos vértices, entre los que se encuentran en la literatura están el algoritmo de búsqueda en profundidad, algoritmo de búsqueda basado en amplitud, y el algoritmo de Dijkstra. De los tres algoritmos, el de Dijkstra garantiza encontrar una ruta y además es la más corta entre dos nodos.

Otra manera de modelar el problema de búsqueda consiste en manejar el entorno como un conjunto de celdas, en la que hay celdas libres, estas celdas son por las que se puede mover el explorador, están conectadas, por lo regular se usa una conectividad cuatro vecinos u ocho vecinos. Con esta representación, se puede elegir una celda libre $O [i, j]$ inicial como punto origen y otra celda $F [i, j]$ como celda final, estas celdas están conectadas, es decir, hay una o varias rutas que permiten llevar de la celda origen a la celda final. El problema consiste en encontrar el camino, conjunto de celdas libres que minimicen el costo de ir de un punto E a otro punto F . El conjunto total de celdas libres, para el entorno que se analiza en el presente trabajo, tiende a incrementarse, dependiendo de la configuración inicial del entorno que se desea crear. Las celdas ocupadas, por el contrario, representan a los obstáculos, y éstas se deben evitar durante el proceso de exploración. Entre los algoritmos que se pueden aplicar a este tipo de

espacios de búsqueda están el A*, que ofrece excelentes resultados para entornos tipo rejilla, otro algoritmo, es una variante del primero, el D*[1].

3. Entorno tipo rejilla

Considere un entorno con una vía para transitarlo. Este entorno, está representado como un arreglo bidimensional con valores “1” y “0”, obstáculos y celdas libres respectivamente. Cada celda tiene vecinos adyacentes en cuatro direcciones Norte, Sur, Este y Oeste.

La forma matricial en “1” y “0”, permite obtener entornos con varias vías, en la figura 1.a se muestra un entorno con dos vías. En la figura 1.b se representa al mismo entorno, pero con distinto número de vías, lo que incrementa el tamaño del entorno.

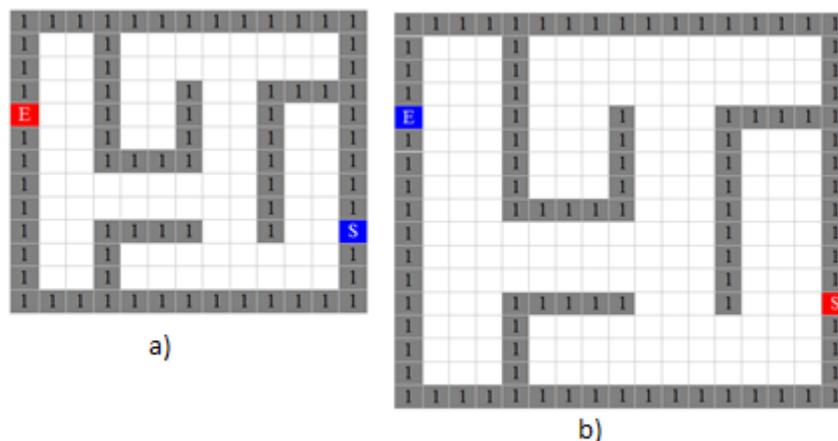


Fig. 1. Entornos ampliados a) con dos vías y b) con tres vías.

Como se puede observar en la figura 2.b, el entorno tiene tres vías en su interior, de manera general, se puede incrementar a n vías, donde n es la cantidad de vías o pasillos que se desea tenga el entorno, algo análogo sería como tener autopistas de dos carriles, tres carriles, etc., es más cómodo para el que maneja por un carril de dos vías que el que maneja en el carril de una sola vía, lo que se debe entender es que a mayor número de vías para navegar por el entorno, involucra tener más movilidad y en términos computacionales tener más control del recorrido del explorador con respecto a la posición que mantiene dentro del entorno. Para profundizar en cómo construir este tipo de entornos, se recomienda consultar [2, 3, 10].

4. Propuesta de etiquetado

El problema que implica encontrar una solución del entorno ampliado es más difícil, ya que hay varios caminos que se pueden recorrer una vez abandonado la celda inicial marcada como entrada, otro problema es, que el explorador puede quedar atrapado en

una sección muy pequeña del entorno, por lo que de alguna manera se debe “recordar” las celdas ya visitadas o transitadas.

Para etiquetar el entorno, se propone considerar las posiciones [i, j] dentro de la matriz en donde se tiene una conectividad cuatro vecinos, en función de ello, asignar una etiqueta a la celda actual para examinar las diferentes celdas adyacentes a las que se puede mover. En la figura 2 se ha indicado con una “s” las celdas que tienen probabilidad de choque *cero*: el explorador tiene libertad de moverse en cualquiera de las cuatro direcciones posibles; en las orillas, celdas marcadas con una “o”, tiene probabilidad de choque del 25%: el explorador puede moverse en tres de las direcciones posibles; en las esquinas, celdas marcadas con una “e”, aumenta la probabilidad de choque en un 50%; el explorador puede moverse en dos de las opciones posibles.

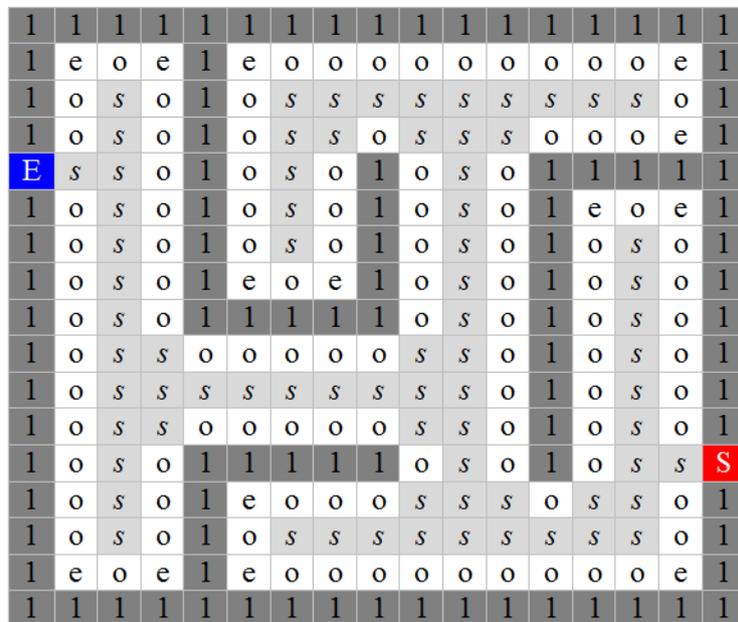


Fig. 2. Entorno etiquetado.

En el espacio de búsqueda de la figura 2, celdas etiquetadas, se observa que se puede reducir el número de celdas libres a explorar una vez abandonado el punto inicial E, es deseable que el recorrido se realice por las celdas etiquetadas como “s”, esto hace que la navegación en el entorno esté libre de colisiones. Otro punto a observar es que las celdas etiquetadas como “s”, están conectadas, es decir, se puede elegir cualquier celda como punto inicial y otra celda como punto final, por lo que se puede ahora encontrar un conjunto de celdas “s” que represente la trayectoria solución entre ambas celdas.

4.1. Etiquetado

El proceso de realizar el etiquetado en el entorno ampliado implica hacer un recorrido del espacio bidimensional de manera a priori, en la que se usan mascarar de tres por

tres con conectividad cuatro vecinos, y con éstas se valida la posición central $p [i, j]$ de la máscara respectiva y validar las celdas adyacentes a los que se puede llegar. Cada máscara permite etiquetar de manera única a una celda.

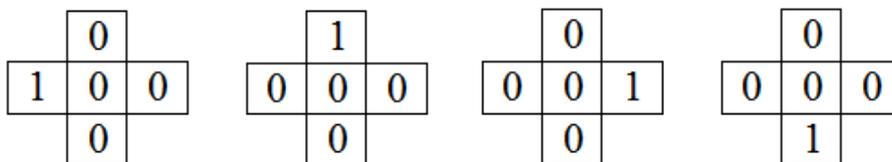


Fig. 3. Máscaras para etiquetar celdas tipo “o”.

En la figura 3, se muestra las máscaras que se utilizan para etiquetar celdas tipo “o”, estas celdas tienen tres celdas libres con una probabilidad de choque del 25%.

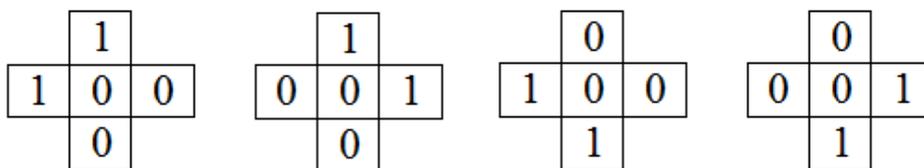


Fig. 4. Máscaras para etiquetar celdas tipo “e”.

En la figura 4, se muestran las máscaras que se utilizan para etiquetar las celdas tipo “e”, estas celdas tienen dos celdas libres con una probabilidad de choque del 50%.

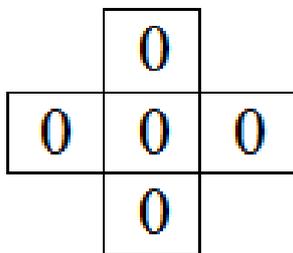


Fig. 5. Máscara para etiquetar celdas tipo “s”.

En la figura 5, se muestra la máscara que se utiliza para etiquetar celdas tipo “s”, estas celdas son las que están libres de obstáculos, tienen cuatro direcciones de movilidad.

4.2. Búsqueda de ruta solución en espacio etiquetado

Para encontrar la trayectoria que conecta a la celda origen E y la celda final S, se ha utilizado el modelo de representación de grafos, en este caso, el total de celdas tipo “s” se le asocia un conjunto de nodos junto con la posición respectiva del entorno.

Esta representación hace suponer que se tienen tantos nodos, como celdas “s” haya en el conjunto de celdas etiquetadas en el entorno, sin embargo, se han discriminado

algunas celdas “s” con la finalidad de construir un grafo menos denso, en la que se aplica el algoritmo de Dijkstra entre dos nodos para encontrar el camino más corto y así visualizar la solución [5, 6].

5. Resultados obtenidos

El entorno a explorar depende del número de vías y por lo tanto influye en el tamaño del entorno virtual. El objetivo se cumplió al permitir reducir el número de posibles celdas a explorar al recorrer el entorno teniendo mayor grado de libertad en los movimientos realizados.

Las salidas que arroja el sistema se guardan en archivos TXT para facilitar la lectura y representación gráfica de los entornos.

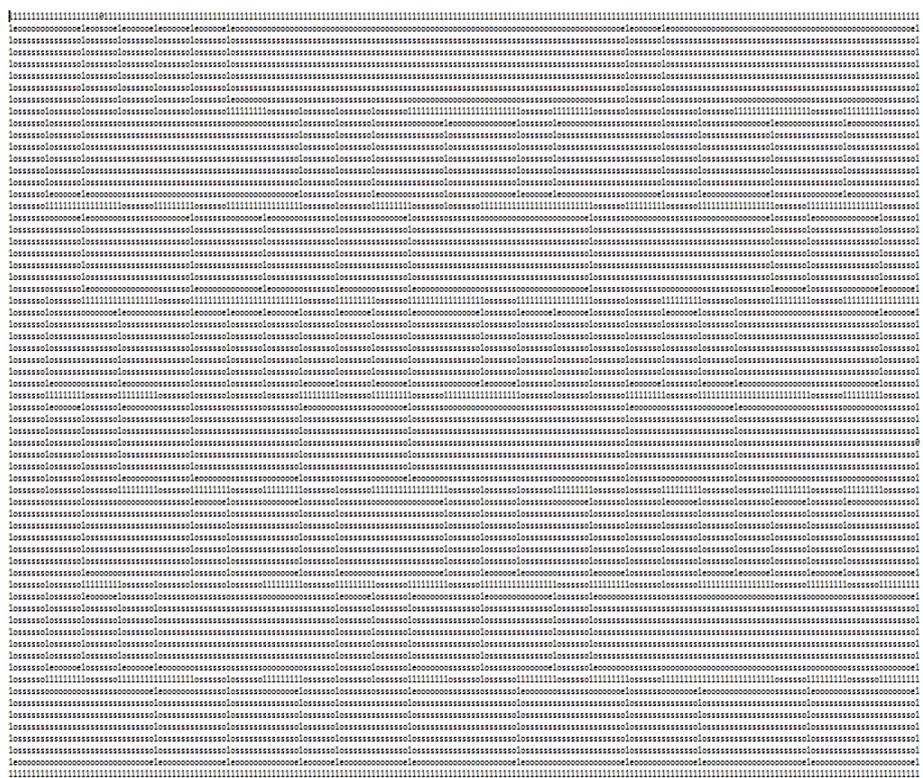


Fig. 6. Entorno con 7 vías.

En la figura 6 se muestra un archivo binario que representa a un entorno con siete vías en su interior, como se aprecia, no es posible hacer un hurgamiento visual de manera cómoda. Sin embargo, se genera un entorno virtual con salida binaria completamente etiquetado.

En la figura 7 se tiene un archivo binario que representa a un entorno con cinco vías de libertad. De los resultados obtenidos, en el 100% se etiqueta el espacio de búsqueda así como lograr reducir el espacio de búsqueda.

Víctor Tomás Tomás Mariano, Felipe de Jesús Núñez Cárdenas, Efraín Andrade Hernández

7. Yandun, A., Sotomayor, N.: Planeación de y seguimiento de trayectorias por un robot móvil. Escuela Politécnica Nacional, Quito, Ecuador (2005)
8. Zalamo, G., Gonzalez, A., Delgado, J., Bermejo, J.: Construcción de Mapas de entorno y estimación de la posición de un robot móvil. Departamento de Ingeniería y Sistemas y automático, Universidad de Valladolid (2006)
9. Ahlquist, J., Novak, J.: Game Development Essentials: Game Artificial Intelligence, Capítulo 6 (2008)
10. Tomas, M.V.T., Pozas, C.M., Hernández, C.J.: Propuesta para la generación de laberintos ampliados en 2D. Simposio Iberoamericano Multidisciplinario de Ciencias e Ingenierías (SIMCI), Zempoala, Hidalgo (2011)

Hacia un modelo de interfaces multimodales adaptables a los canales de aprendizaje en aplicaciones colaborativas como apoyo a la educación

Itzel Alessandra Reyes Flores¹, Carmen Mezura-Godoy¹, Gabriela Sánchez Morales²

¹ Universidad Veracruzana, Xalapa, Veracruz,
México

² CONACYT- Universidad Veracruzana, Xalapa, Veracruz,
México

zS15019630@estudiantes.uv.mx, cmezura@uv.mx, gsanchezmo@conacyt.mx

Resumen. Las interfaces multimodales permiten que la interacción humano-computadora se dé en forma natural, y que la unión entre el mundo real con el digital sea totalmente transparente para el usuario. Actualmente, existen aplicaciones educativas que utilizan este tipo de interfaces; sin embargo, aún existen problemas con estos sistemas, por ejemplo, no consideran el trabajo en equipo, la adaptación al usuario es pobre, y existen pocos modelos que apoyen su desarrollo. Por tal motivo en este trabajo se muestra una versión preliminar de un modelo que considera las interfaces multimodales en aplicaciones colaborativas como apoyo en la educación, tomando en cuenta la adaptación al contexto como son los canales de aprendizaje de los usuarios, el ruido ambiental y los dispositivos en uso. Posteriormente se muestra una arquitectura para dar un panorama general de cómo se llevará a cabo la adaptación mediante mecanismos de inferencia.

Palabras clave: Interfaces multimodales, aplicaciones educativas, aplicaciones colaborativas, canales de aprendizaje, mecanismos de inferencia.

Towards a Model of Adaptable Multimodal Interfaces Considering Learning Channels to support education

Abstract. Multimodal interfaces allow a smooth human-computer interaction, and connection between real and digital worlds to be entirely transparent to the user. Most current educational applications using this kind of interfaces does not support collaboration and they are not adaptable. Furthermore, few models have been developed for development of these applications. In this paper, we propose a preliminary model of multimodal interfaces for educational collaborative applications. Moreover, a corresponding architecture is also proposed in order to give a general overview on how adaptation is supported by inference techniques.

Keywords: Multimodal interfaces, educational applications, groupware, learning channels, inference mechanisms.

1. Introducción

La investigación y el desarrollo de Interfaces de Usuario Multimodales (IUM) ha ido creciendo en los últimos años por la necesidad de tener sistemas que interactúen con el usuario de forma natural y transparente. Las IUM han demostrado tener muchas ventajas comparadas con interfaces mono modales, ya que las primeras facilitan el acceso a la información y a la funcionalidad del sistema a través de una gama de diferentes canales de entrada y de salida [1], evitan los errores, proveen robustez a la interfaz, ayudan al usuario a corregir errores o recuperarse de ellos más fácilmente, y añaden métodos de comunicación alternativos a diferentes situaciones y ambientes [2].

Dentro de las aplicaciones que consideran IUM, se encuentran, NUILAC, que posibilita interactuar con un *Learning Management System* como Moodle [3]; PSYBENCH permite que los usuarios geográficamente distantes puedan colaborar en un espacio de trabajo compartido utilizando interfaces tangibles [4], AIR Conferencing que mediante grabaciones de video y conversiones de voz a texto el usuario puede repetir lo que se perdió al estar ausente de la conferencia [5], y Thunderwire genera un espacio compartido entre usuarios conectados utilizando únicamente audio [6].

Las IUM responden a la entrada de más de una modalidad (entrada táctil, reconocimiento de voz, salida audiovisual de voz, y otros), utilizando diferentes dispositivos de entrada y salida de la computadora, haciendo uso de diferentes canales sensoriales [7]. Por otro lado, se dice que la modalidad está estrechamente unida con los sentidos humanos. Raisamo [8] distingue tres modalidades (Visual, Auditiva, y Táctil) que están relacionadas con los canales de aprendizaje.

Los canales de aprendizaje son el conjunto de estrategias, métodos y tendencias concretas que se utilizan cuando se quiere aprender algo [9]. De acuerdo a [11], toda persona percibe la información a través de canales de aprendizaje, es decir, por medio de los ojos (canal visual), de los oídos (canal auditivo) y de sensaciones del cuerpo (canal kinestésico). Sin embargo, son muy pocas las aplicaciones que consideran esta característica para beneficiar la interacción del usuario con su interfaz, y en menor medida consideran la colaboración grupal. Dicha colaboración es uno de los objetivos principales de las aplicaciones colaborativas.

Las aplicaciones colaborativas o groupware, como normalmente se les llama en la *Computer-Supported Cooperative Work*, son aplicaciones que apoyan el trabajo en equipo [1], cuyo objetivo es ayudar a los grupos en la comunicación, la colaboración y la coordinación de sus actividades, trabajando en entornos comunes y virtuales [12]. Este tipo de aplicaciones a menudo son utilizadas en contextos educativos, permitiendo la participación activa y la interacción entre estudiantes y profesores que comparten sus ideas e información para lograr un objetivo. Google, por ejemplo, pone a disposición de los centros educativos de forma gratuita Google Apps para educación, que es un conjunto de herramientas como correo Gmail, Google Calendar, Google Drive, entre otros. Por otro lado, se han desarrollado aplicaciones como Mindmeister, que permite la colaboración entre estudiantes en línea para la elaboración colaborativa de mapas mentales que se pueden editar sucesivamente [13].

Aun cuando ya existe una gran cantidad de aplicaciones colaborativas que apoyan a la educación, aún existen desafíos en la interacción de los estudiantes con estas aplicaciones que imposibilitan su aprovechamiento total; por ejemplo, mostrar la misma interfaz para todos los participantes sin considerar algo tan importante como son los canales de aprendizaje. Esto hace que la interacción se dé utilizando todos los sentidos de manera general, permitiendo así, la posibilidad de que el estudiante presente distracciones en el proceso de aprendizaje y colaboración.

Es importante destacar que actualmente existen aplicaciones colaborativas multimodales como son, AwarenessMaps, diseñado para compartir información entre los usuarios mediante una serie de imágenes con actividades recientes, y AudioSpace, una interfaz de audio que permite a los usuarios especificar contextos que les interesa [14]. Aun cuando estas aplicaciones consideran la multimodalidad la mayoría no se adaptan al contexto, es decir, están hechas sin considerar las situaciones en el ambiente externo que pueden afectar la interacción multimodal; por ejemplo, un usuario interactúa con un sistema mediante las modalidades de audio y reconocimiento de voz. Sin embargo, el sistema no considera que esta interacción puede afectar la concentración de otras personas al rededor; tampoco considera si el ambiente externo es muy ruidoso, o si los dispositivos de entrada y salida necesarios para la interacción están conectados y disponibles.

Dado el preámbulo anterior, en este trabajo se muestra una versión preliminar de un modelo que considera las interfaces multimodales en aplicaciones colaborativas como apoyo a la educación, tomando en cuenta los canales de aprendizaje, el ambiente externo y los dispositivos de entrada y salida. Este artículo se organiza de la siguiente manera, en la sección 2 se presentan los modelos, *frameworks* y arquitecturas importantes para este tema de investigación; en la sección 3, el modelo propuesto y en la sección 4 se muestra la arquitectura, los mecanismos de inferencia para la adaptación y un escenario para ejemplificar el uso de la arquitectura. Finalmente, en la sección 5 se presentan las conclusiones y el trabajo a futuro.

2. Estado del arte

Existen pocos modelos de interfaces multimodales que consideran contextos educativos, sin embargo, ninguno está enfocado en aplicaciones colaborativas, ni toman en cuenta los canales de aprendizaje del usuario. Por lo que a continuación se describen modelos, *frameworks* y arquitecturas que se han desarrollado hasta hoy, pero que contemplan cada una de estas características por separado.

Algunas investigaciones se han enfocado en diseñar *frameworks* y arquitecturas para el desarrollo de interfaces multimodales; por ejemplo, el trabajo de Flipplo et al. [15] provee un *framework* para el desarrollo rápido de estas interfaces, en las que los usuarios tienen la libertad de elegir y combinar la interacción con el sistema y así crear una forma más eficiente y agradable de trabajo. Dynamo [16] una arquitectura para diseñar interfaces multimodales adaptables a entornos generalizados, basados en servicios; esta arquitectura toma en cuenta la plataforma, el peso de los componentes y un gestor autónomo basado en modelos para construir y mantener la interacción multimodal en tiempo de ejecución.

AMENITIES es una arquitectura centrada en el concepto de grupo basada en modelos de comportamiento y tareas para el análisis, diseño y desarrollo de sistemas cooperativos [17]. Y el modelo de Jun et al [18] para *E-learning*, basado en la tecnología que permite el trabajo en grupo, en el que se utilizan redes para mejorar la eficiencia del aprendizaje, la creación y el intercambio de conocimientos.

En la investigación y desarrollo de sistemas educativos también se han comenzado a desarrollar modelos. Uno de ellos es el modelo de Silva et al [19] basado en técnicas instruccionales y estilos de aprendizaje para el apoyo en la creación de Objetos de Aprendizaje (OA). Este modelo permite al profesor, a través del uso de una plataforma tecnológica, definir los objetivos de aprendizaje y los procesos cognitivos implicados. De igual forma el profesor define el estilo de aprendizaje de los estudiantes a quien va dirigido el OA, el modelo evalúa 36 técnicas instruccionales; posteriormente, selecciona las más adecuadas a los procesos cognitivos y a los estilos de aprendizaje utilizando un modelo matemático.

Tabla 1. Modelos, *frameworks* y arquitecturas desarrolladas actualmente.

Nombre	Enfoque	Colaborativo	Multimodal	Adaptable al usuario	Aplicación desarrollada
Framework de Flippo et al (2003).	General	No	Señalamiento por dispositivo y <i>Gaze</i>	No	Flatscape (Aplicación Colaborativa)
Arquitectura Dynamo (2012).	General	No	Señalamiento con dispositivo	No	No ³
Arquitectura AMENITIES (2004).	General	Si	No	No	Agenda Colaborativa
Modelo de Jun et al (2011).	Educativo	Si	No	No	No
Modelo de Silva et al (2012).	Educativo	No	No	Estilos de Aprendizaje, Objetivos de Aprendizaje.	Plataforma Tecnológica
Modelo de Luna (2016).	Educativo	No	No	Canales de Aprendizaje, Estilo de Aprendizaje, Motivación, Cultura, Limitación, Modelo mental, Estrategia de Aprendizaje.	No ³

³ Se describen escenarios en lugar de desarrollarse alguna aplicación utilizando este modelo o arquitectura.

Otro modelo que considera contextos educativos y canales de aprendizaje es el propuesto por Luna [11], para el desarrollo de aplicaciones interactivas en el que conceptualiza las características propias del usuario, estilos, canales de aprendizaje y su experiencia en el uso de sistemas educativos.

En la Tabla 1 se comparan los trabajos anteriormente mencionados, destacando principalmente su enfoque y el desarrollo de aplicaciones colaborativas; además si consideran la multimodalidad, al usuario y el desarrollo de alguna aplicación.

En la Tabla 1 se observa que dos de los seis trabajos toman en cuenta la multimodalidad, y con uno de ellos se desarrolló una aplicación colaborativa como caso de estudio. Sólo dos se enfocan en el desarrollo de aplicaciones colaborativas, y tres en el desarrollo de aplicaciones educativas que se adaptan al usuario; por ejemplo, consideran el estilo de aprendizaje, objetivos de aprendizaje, cultura etc.

Aun cuando estos trabajos de investigación consideran contextos educativos, aplicaciones colaborativas, interfaces multimodales y la adaptación del usuario, ninguno contempla unir estos elementos para enriquecer la interacción del usuario, como se propone en este trabajo. Es importante mencionar que el modelo que más se asemeja a esta propuesta es el de Luna [11], solo que este no considera las modalidades ni tampoco el trabajo colaborativo. Por tal motivo, se contempla la necesidad de crear un modelo que utilice interfaces multimodales adaptables a los canales de aprendizaje en aplicaciones colaborativas como apoyo a la educación.

3. Propuesta del modelo

En esta sección se describe una versión preliminar del modelo de interfaces multimodales adaptables a los canales de aprendizaje en aplicaciones colaborativas como apoyo a la educación. Este modelo considera ocho elementos: **usuario, canal de aprendizaje, dispositivos de interacción, equipo heterogéneo, rol, actividad, ambiente externo, y aplicación colaborativa**; y dos subelementos dentro de la actividad: **tareas cooperativas y tareas colaborativas**. En la Fig. 1, pueden observarse los elementos, subelementos y las relaciones entre ellos.

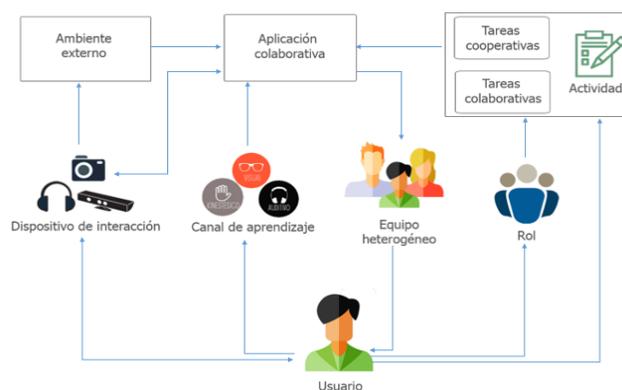


Fig. 1. Modelo de interfaces multimodales adaptables a los canales de aprendizaje en aplicaciones colaborativas como apoyo a la educación.

3.1. Elementos

El elemento **usuario** es la persona física que utiliza la aplicación, ya sea el profesor o el alumno.

El **canal de aprendizaje** se refiere al modo en el que el usuario percibe la información. Usualmente predomina un solo canal en el alumno. Se consideran tres canales de aprendizaje: visual, auditivo y kinestésico.

Los **dispositivos de interacción** son los que permiten la comunicación entre el usuario con la aplicación. Éstos pueden ser de entrada, de salida o ambas.

El **ambiente externo** es la información que se obtiene de lo que pasa alrededor de la aplicación. Específicamente sobre los dispositivos conectados o desconectados a la computadora, la cantidad de personas que se encuentren alrededor del usuario y el ruido del ambiente.

Un **equipo heterogéneo** es un grupo de trabajo conformado por usuarios con diferentes canales de aprendizaje. En un caso ideal cada equipo tendrá un integrante de cada canal (visual, auditivo y kinestésico); de otra forma, se procurará formar equipos con al menos dos tipos de canales de aprendizaje diferentes.

Un **rol** es el papel que desarrolla el usuario dentro de la actividad colaborativa.

La **actividad** es el conjunto de operaciones que se llevan a cabo para cumplir un objetivo. Ésta consiste en la ejecución de ciertas tareas mediante la utilización de diversos recursos.

La **aplicación colaborativa** se refiere a los métodos que facilitan el trabajo en grupo, para este tema de investigación se quiere apoyar a los usuarios localizados en diferentes puntos geográficos a que trabajen en un objetivo en común. En la sección 4 se describe la arquitectura de éste elemento.

La actividad está conformada por **tareas cooperativas** y **tareas colaborativas**. En una **tarea cooperativa** los alumnos se dividen en sub tareas y cada quien se hará responsable de cumplir con su parte. En cambio, en la **tarea colaborativa** todos pueden trabajar en todas las sub tareas

3.2. Relaciones

Cada elemento del modelo está relacionado con otro. El **usuario** tiene un **canal de aprendizaje**, un **dispositivo de interacción**, un **rol**, una **actividad** y es parte de un **equipo heterogéneo**.

La **aplicación colaborativa** recibe como entrada el **canal de aprendizaje** que predomina en el usuario y mediante mecanismos de inferencia podrá asignar los **equipos heterogéneos**. Además, recibe la información del **ambiente externo**, y las **tareas** asignadas a cada integrante del equipo; esto, junto con el canal de aprendizaje, le permitirán activar, también mediante mecanismos de inferencia, la modalidad con la que va a interactuar el usuario con la interfaz, y los componentes con los que será conformada.

El **rol** del usuario intervendrá en la asignación de **actividades**. Cuando el usuario tiene un rol de profesor, este podrá asignar una actividad, y cuando sea un alumno será quien realice las **tareas cooperativas** o las **tareas colaborativas** en las que se dividió la actividad, asignándole a cada usuario una tarea y con ello un nuevo rol.

4. Propuesta de la arquitectura

Con base en el modelo propuesto se describe una arquitectura de uno de sus elementos: **aplicación colaborativa**. En esta arquitectura se toman en cuenta cuatro elementos importantes: **interfaz de usuario, adaptación, lógica del negocio y comunicación entre dispositivos**. En la Fig. 2 se pueden identificar estos elementos y la relación que existe entre cada uno.

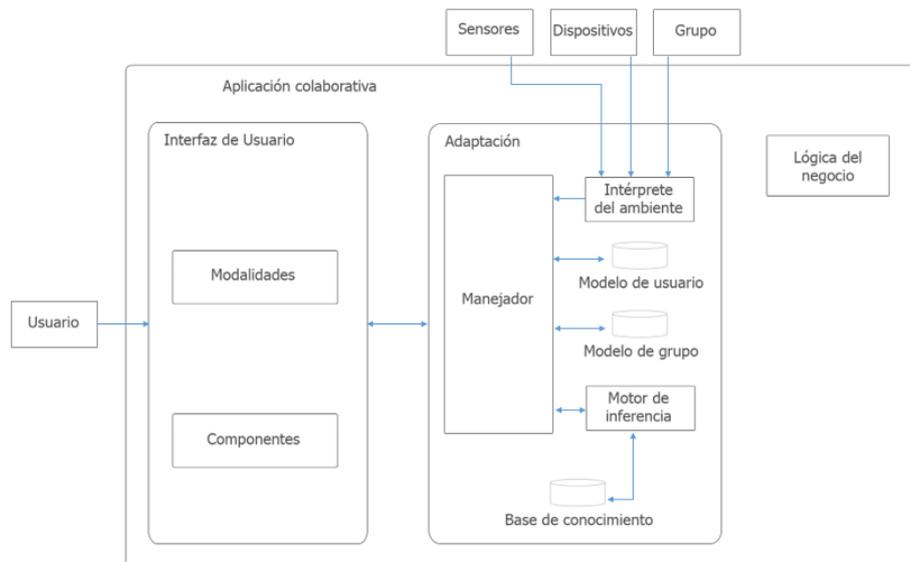


Fig. 2. Arquitectura de la aplicación colaborativa.

La **interfaz de usuario** es el medio de interacción entre el usuario y la aplicación. Por lo que la interfaz recibe como entrada el **usuario** que interactúa en ese momento con ésta. Dentro de la interfaz de usuario se consideran dos elementos; las **modalidades** de interacción, que significan el total de modalidades con el que el usuario podría interactuar con la aplicación. Éstas pueden ser reconocimiento de voz, audio, reconocimiento de la mirada, reconocimiento de gestos, entre otros. Así mismo se consideran los **componentes** que serán los elementos que conformarán la interfaz, ya sean botones, menús, barras y otros.

La **adaptación** le da a la aplicación, la capacidad de aprender y cambiar considerando la información que recibe del ambiente externo mediante los **sensores**, los **dispositivos de interacción** y un **grupo** determinado que se busque distribuir en equipos más pequeños de forma heterogénea. Dicha capacidad de aprender se realizará con mecanismos de inferencia. En específico la adaptación activará la modalidad con la que el usuario va a interactuar con la interfaz, y asignará los equipos heterogéneos de un grupo determinado.

La información obtenida por el ambiente externo es encapsulada por el **intérprete del ambiente (IA)** y si es necesario lo traduce a otros formatos; por ejemplo, si alguna función de la aplicación utiliza un sensor de ruido, pero éste da resultados de un tipo

diferente al que recibe como parámetros la función, es el IA el que se encarga de la traducción.

Así mismo, el IA se comunica con el **manejador**. Éste manejador utiliza los datos del intérprete para saber las acciones a realizar (hacer consultas, llamar funciones, generar nuevos hechos, entre otros). Además, el manejador guarda y recupera información del **modelo de usuario** y/o del **modelo de grupo**. El modelo de usuario almacena un conjunto de datos, de manera estructurada, que utiliza la aplicación para respaldar información acerca del usuario que utiliza la aplicación en ese momento. Y el modelo de grupo almacena los datos del grupo determinado, del cual se requieren formar equipos heterogéneos.

Para obtener resultados acerca de la modalidad a activar o de los equipos heterogéneos que habrá que conformar, el manejador le envía a un **motor de inferencia** la información del modelo de usuario y/o del modelo de grupo. Éste motor de inferencia, a su vez consulta una **base de conocimientos** que contiene hechos que se consideran ciertos e incuestionables, y un conjunto de reglas conformadas por premisas o términos y una conclusión. El motor consulta la lista de hechos y verifica si con ellas se cumplen todos los requisitos o premisas de una o más reglas; si esto es cierto, entonces se verifica que a la conclusión que se llega está dentro de la lista de hechos, sino está, entonces se agrega como uno nuevo, así la aplicación aprende nuevo conocimiento hasta llegar a una conclusión final. En este caso, activar la modalidad con la que el usuario va a interactuar con la interfaz o la cantidad de equipos heterogéneos conformados y cómo estarán distribuidos para realizar la actividad colaborativa.

Así mismo, dentro de la aplicación colaborativa, se considera la **lógica del negocio** que es el núcleo funcional de la aplicación, es decir, debe contener todos los requerimientos necesarios dependiendo la aplicación colaborativa, por ejemplo, editor de textos, lluvia de ideas, entre otros. Además, se encarga de la lógica detrás de la interfaz de usuario; pueden ser, entradas y consultas de datos, procesamiento de información y entrega de resultados al usuario.

4.1. Escenario

En el siguiente escenario se considera los elementos de la arquitectura, por tal motivo se encontrarán dichos elementos a lo largo de la descripción, así como las entradas a la aplicación colaborativa.

En un salón de clases se imparte la materia de lectura y redacción (*grupo*) a 10 estudiantes de licenciatura en informática. Durante la clase el profesor les solicita a los alumnos que realicen una presentación (*actividad*) en diapositivas para exponer sobre las reglas gramaticales. Esta actividad será en equipo, lo cual el profesor lo indica en la aplicación colaborativa de apoyo. Entonces la aplicación recupera la información de los estudiantes (**modelo de usuario**) para sugerir como organizarlos en equipos (**motor de inferencia**). De los 10 estudiantes se encuentran 7 con un canal de aprendizaje visual, 2 con auditivo y 1 con kinestésico, información que la aplicación colaborativa previamente conoce, por lo que propone tres opciones: 5 equipos conformados por 2 integrantes, 2 equipos formados por 5 integrantes o 3 equipos formados con diferente número de estudiantes (2 equipos de 3 personas y 1 equipo de 4 personas). Por ejemplo, en la primera opción tiene 2 equipos integrados con un alumno visual y un auditivo, 1

equipo integrado con un alumno kinestésico y un visual, y 2 equipos integrados con dos alumnos visuales respectivamente. En este caso el profesor selecciona la opción uno.

El profesor ingresa a la aplicación, asigna la actividad, y la divide en sub tareas cooperativas. Dependiendo el rol de cada integrante del equipo (*equipo heterogéneo*) la aplicación asignará una sub tarea a cada integrante, y de acuerdo a su canal de aprendizaje designará la modalidad de interacción y los componentes de la interfaz.

Cada uno de los integrantes del equipo tienen que cumplir con la actividad fuera de la clase, así que ellos prefieren hacerla en casa, por lo cual cada uno de ellos ingresa a la aplicación desde su computadora. Supongamos que el alumno Pedro, tiene un canal auditivo, como tarea le corresponde escribir los títulos y subtítulos de cada diapositiva, y revisar la información que el resto de sus compañeros haya escrito. Tiene conectados sus audífonos con micrófono integrado (*dispositivos*), y se encuentra sólo en una habitación sin ruido (*sensores*), por lo que la aplicación le ha asignado las modalidades de reconocimiento de voz y audio (**motor de inferencia**).

5. Conclusiones y trabajo futuro

En este artículo se propone una versión preliminar de un modelo y una arquitectura que consideran las interfaces multimodales en aplicaciones colaborativas como apoyo a la educación, tomando en cuenta la adaptación al usuario, en específico a sus canales de aprendizaje. Como se habló en la sección de estado del arte, existen trabajos relacionados con estas características. Cabe mencionar que de los seis trabajos identificados sólo tres de estos tienen un enfoque educativo, como lo hace también esta investigación.

De igual forma, sólo dos de estos trabajos relacionados proponen interfaces utilizando modalidades como, señalamiento por dispositivo y *Gaze*. A diferencia de estos trabajos, el modelo planteado en esta investigación toma en cuenta multimodalidades relacionadas con los canales de aprendizaje; reconocimiento de voz, salida de audio (canal auditivo), salida visual, señalamiento por dispositivo (canal visual), y reconocimiento de gestos (canal kinestésico).

Por otra parte, como resultado de la investigación de la literatura, se identificó una arquitectura y un modelo que sirven de referencia para el desarrollo de aplicaciones colaborativas. Sin embargo, estos trabajos no incluyen interfaces multimodales como lo propone este trabajo, y no se adaptan al usuario.

Así mismo, cabe mencionar que en el estado del arte se identificaron dos modelos que toman en cuenta la adaptación al usuario, considerando características propias como, el estilo de aprendizaje, los objetivos de aprendizaje, los canales de aprendizaje, la motivación, la cultura, la limitación, el modelo mental, y la estrategia de aprendizaje. Sin embargo, el modelo propuesto en este trabajo contempla la adaptación al usuario considerando los canales de aprendizaje, el ambiente externo en el que el usuario se encuentra, los dispositivos de entrada y salida conectados al momento de interactuar con la aplicación, y las actividades que le fueron asignadas para determinar la modalidad de interacción entre el usuario y la interfaz. En la arquitectura propuesta se presentan los mecanismos de inferencia para adaptar la interfaz respecto a la modalidad que se activará dado el contexto del usuario y del ambiente. Además, se propone otro mecanismo de inferencia para distribuir de manera heterogénea los equipos de trabajo

que utilizarán la aplicación, con el fin de que exista, en el mejor de los casos, un integrante de cada canal de aprendizaje (visual, auditivo, kinestésico).

Dado a que este trabajo de investigación aún se encuentra en desarrollo, como trabajo futuro, se desarrollarán los mecanismos de inferencia para asignar grupos heterogéneos de trabajo, para adaptar la interfaz y las modalidades de interacción. El modelo y la arquitectura planteada se probarán mediante un caso de estudio que consistirá en desarrollar una aplicación colaborativa como apoyo a la educación que utilice interfaces multimodales para la interacción con el alumno y se adapte de acuerdo a su canal de aprendizaje; por ejemplo, activar modalidades de reconocimiento de voz y audio para un alumno con canal auditivo, y adaptar la interfaz de usuario con un lector de pantalla. Actualmente, dicha aplicación está en la fase de análisis de requerimientos, en la cual se están incluyendo alumnos de la licenciatura en Informática de la Universidad Veracruzana. Además, como otro de los trabajos a futuro es el enriquecimiento del modelo y arquitectura propuesta, con el fin de considerar los elementos faltantes o sobrantes que se detecten al desarrollar la aplicación colaborativa.

Agradecimientos. Agradecemos a CONACYT por todos los apoyos, en específico por la beca con número 590658, así como a la Universidad Veracruzana por el apoyo brindado para el desarrollo de esta investigación.

Referencias

1. Dix, A., Finlay, J., Abowd, G., Beale, R.: Human-Computer Interaction. Pearson Education Limited. pp. 366–370, 663–665 (1993)
2. Jaimes, A., Sebe, N.: Multimodal human-computer interaction: A survey. *Computer vision and image understanding*. Vol. 108, No. 1, pp. 116–134 (2007)
3. Vásquez, A., Duque, E.: Nui para la educación, eliminando la discriminación tecnológica en la búsqueda de la inclusión digital. Corporación Universitaria Americana (2013)
4. Brave, S., Ishii, H., Dahley, A.: Tangible Interfaces for Remote Collaboration and Communication. Proceedings of the 1998 ACM conference on Computer supported cooperative work, ACM, pp. 169–178 (1998)
5. Inkpen, K., Hegde, R., Junuzovic, S., Brooks, C., Tang, J.C.: AIR Conferencing: Accelerated Instant Replay for In-Meeting Multimodal Review. Proceedings of the 18th ACM international conference on Multimedia, ACM, pp. 663–666 (2010)
6. Ackerman, M., Starr, B., Hindus, D., Mainwaring, S.: Hanging on the 'Wire: A Field Study of an Audio-Only Media Space. *ACM Transactions on Computer-Human Interaction*, Vol. 4, No. 1, pp. 39–66 (1997)
7. Möller, S., Engelbrecht, K., Kühnel, C., Wechsung, I., Weiss, B.: A taxonomy of Quality of Service and Quality of Experience of Multimodal Human-Machine Interaction. *Quality of Multimedia Experience, QoMEx, International Workshop on. IEEE*, pp. 7–12 (2009)
8. Raisamo, R.: Multimodal Human-Computer Interaction: a constructive and empirical study. Tampereen yliopisto (1999)
9. Tomas, U.: Canales del Aprendizaje. El psicoasesor. <http://elpsicoasesor.com/canales-del-aprendizaje/> (2011)
10. Valdivia, J.: El conocimiento de los estilos de aprendizaje como medida de atención a la diversidad y sus implicaciones educativas en educación infantil. *Revista digital enfoques educativos*, Vol. 75, pp. 85–94 (2011)
11. Luna, J.: Modelo de interacciones para el desarrollo de aplicaciones interactivas en contextos educativos. Universidad Veracruzana (2016)

12. Ellis, C.A., Gibbs, S.J., Rein, G.L.: Groupware some issues and experiences. *Communications of the ACM*, Vol. 34, No.1, pp. 38–58 (1991)
13. Abdul, J., Velarded, R.: Herramientas Web 2.0 para el Aprendizaje Colaborativo. http://remo.det.uvigo.es/solite/attachments/038_Web%202.0.pdf (2009)
14. Gross, T.: Universal Access to Groupware with Multimodal Interfaces. *Proc. of the second Int. Conf. on Universal Access in HCI: Inclusive Design in the Information Society*, pp. 1108–1112 (2003)
15. Flippo, F., Krebs, A., Marsic, I.: A Framework for Rapid Development of Multimodal Interfaces. *Proceedings of the 5th international conference on Multimodal interfaces*, ACM, pp. 109–116 (2003)
16. Avouac, P.A., Lalanda, P., Nigay, L.: Adaptable multimodal interfaces in pervasive environments. *Consumer Communications and Networking Conference. IEEE*, pp. 544–548 (2012)
17. Garrido, J.L., Gea, M., Noguera, M., González, M., Ibáñez, J.A.: Una Propuesta Arquitectónica para el Desarrollo de Aplicaciones Colaborativas. *Interacción*, pp. 164–171 (2004)
18. Jun, Y., Suicheng, L., Rui, H.: The model of Networked Learning Environment for E-learning Based on Groupware Technology. *Computer Science & Education (ICCSE), 6th International Conference on*, Vol. 8, No. 11, pp. 995–998 (2011)
19. Silva, A., Ponce, J.C., Villalpando, M.D.: Modelo para la Creación y Uso de Objetos de Aprendizaje, Basado en la Valoración de Técnicas Instruccionales. *Conferencias LACLO*, Vol. 3, No. 1 (2012)

Una base de conocimientos para asistir el aprendizaje de la programación

Juan P. Ucán, Antonio A. Aguilera, Raúl A. Aguilar

Universidad Autónoma de Yucatán, Facultad de Matemáticas,
México

{juan.ucan, aaguilet, avera}@correo.uady.mx

Resumen. Con el propósito de ampliar la investigación en el ámbito del aprendizaje de la programación, se ha realizado un trabajo de investigación que involucra, como estrategia de análisis de nuevas opciones instruccionales, el desarrollo de un Entorno Virtual Colaborativo Inteligente (EVCI). En este artículo se presenta la arquitectura del EVCI, en particular, se describe la Base de Conocimientos del Sistema Experto, la cual es integrada a manera de componente al EVC. Las principales funcionalidades del Sistema Experto, y por tanto prestaciones del EVCI son también descritas en el artículo.

Palabras clave: EVC, EVCI, sistema experto.

A Knowledge Base to Assist the Learning of the Programming

Abstract. In order to expand research about the learning of the programming, it has done a research that involved as strategy of analysis about new instructional options, development of an Intelligent Collaborative Virtual Environment (EVCI). In this paper we describe an EVCI architecture, particularly the Knowledge Base of Expert System, which is integrated to EVC as a component. The main features of Expert System and EVCI are also described in the paper.

Key Words: EVC, EVCI, expert system.

1. Introducción

El aprendizaje de la programación, a nuestros días aún no resulta una tarea resuelta en el ámbito de la Educación en Informática. El primer lenguaje seleccionado, el primer paradigma abordado, los errores comunes cometidos por los aprendices, así como las actividades y habilidades cognitivas —independientes del lenguaje y/o paradigma— propias del dominio, son algunos de los factores inmersos en dicha actividad educativa. En este artículo se aborda dicha problemática desde una línea de investigación e

innovación conocida como Entornos Virtuales Inteligentes, en particular, Entornos de Aprendizaje basados en la Colaboración y el Conocimiento.

El trabajo reportado se enmarca en un proyecto de investigación en el que estableció como pregunta de investigación, la siguiente:

“...¿Es posible implementar un Modelo para Asistir en el Aprendizaje de la Programación mediante un Entorno Virtual Colaborativo Inteligente, que facilite al aprendiz la detección de errores comunes de la programación?”[1].

Para dar respuesta a la pregunta de investigación, se propuso el desarrollo de un Marco de Trabajo para Asistir el Aprendizaje de la Programación en Espacios de Trabajo Compartidos y experimentar con dicha propuesta utilizando un Entorno Virtual Inteligente (EVI) diseñado exprofeso. En el presente artículo se describe de manera específica el “Componente Inteligente” diseñado e incorporado al Entorno Virtual de Aprendizaje, dicho componente se corresponde con un Sistema Experto Basado en Reglas que utiliza un motor de inferencia instituido en los errores comunes de la programación.

2. Antecedentes

2.1. Aprendizaje de la programación

La Programación no es una disciplina fácil de aprender y mucho menos de enseñar, ha resultado compleja en su aprendizaje debido a la variedad de sub-tareas y tipos de conocimiento especializados que involucra, y que son necesarios de ejecutar de manera efectiva [2], por otro lado, es una disciplina que debe ser aprendida bajo un enfoque constructivista, es decir, en buena medida se aprende haciendo, y en muchas ocasiones, requiere de un enfoque por aproximaciones sucesivas para poder generar aprendizajes significativos.

En el caso de su enseñanza, la selección del primer lenguaje de programación [3] ha generado discusiones continuas entre los especialistas; vinculado con el lenguaje, el paradigma abordado en los primeros cursos de Informática también ha generado posiciones encontradas, no obstante, y sin un análisis suficientemente discutido, la aparición de enfoques orientados a objetos ha venido desplazando al paradigma estructurado; otros autores por su parte, se han enfocado al aspecto didáctico, y han propuesto algunos modelos para enseñar a programar [4].

Independientemente del proceso educativo a desarrollar, la Programación, como actividad socio-técnica, involucra actividades dependientes del lenguaje, las cuales requieren del conocimiento de la sintaxis, semántica y de la pragmática del lenguaje de programación [5]; por otro lado, requiere de un amplio rango de actividades vinculadas con el proceso de desarrollo software: análisis, diseño, codificación, comprensión de programas, depuración, mantenimiento y documentación.

En el ámbito de la Informática Educativa se han desarrollado diversos sistemas software para asistir el aprendizaje y/o enseñanza de la Programación [6], [7] y [8], sin embargo, para las actividades vinculadas con la depuración y comprensión de programas, tareas a las que no siempre se les otorga la importancia debida, los sistemas software desarrollados, generalmente se orientan al entendimiento del lenguaje, y resultan poco útiles para el entrenamiento de programadores novatos.

2.2. Errores comunes en programación

De acuerdo con el estándar IEEE 610.12-1990 [9], un fallo en el sistema ocurre cuando la salida generada por un programa no coincide con los requisitos establecidos en el proyecto. De ahí que un fallo puede ser resultado de una falta, la cual representa un estado en tiempo de ejecución de un programa, que parece ser incorrecto. Las faltas ocurren como resultado de los errores, los cuales son fragmentos del programa que no cumplen con las especificaciones documentadas, o con las especificaciones del modelo mental de los programadores.

Es importante mencionar, para efectos de la tarea de depuración, que aunque los fallos en la ejecución de los programas garantizan que una o más faltas se encuentran en el código —ya sea por omisión o por comisión del programador—, y que las faltas garantizan que uno o más errores existen, los errores no siempre causan faltas, y las faltas no siempre generan fallos.

En la literatura es posible encontrar una variedad de clasificaciones sobre tipos y causas de errores que pueden ser realizados, principalmente por programadores novatos; dichas clasificaciones han sido generadas a raíz de estudios empíricos sobre patrones encontrados en la actividad de programación. En [10] por ejemplo, se llegó a la conclusión de que una causa de error en la programación, se debe a que los programadores se forman modelos mentales de las especificaciones documentadas, no obstante, dichos modelos resultan insuficientes por las especificaciones inadecuadas, o por el desconocimiento del dominio de la tarea a ser implementada.

Un segundo tipo de errores comunes está relacionado con la sintaxis propia de algunos lenguajes, por ejemplo: el clásico “;” al final de una sentencia en C, la falta de una “}” para delimitar un ciclo o una función, la confusión de los símbolos para realizar asignaciones en lugar de comparaciones, etc.

<code>if (num1 == (num2+num3))....</code>	<i>Correcto</i>
<code>if (num1 = (num2+num3))....</code>	<i>Incorrecto</i>

Otros errores tienen relación con la lógica de las estructuras algorítmicas, en estos, es común encontrarse en las estructuras repetitivas con errores en el número de iteraciones deseadas, p.ej., si se desea ejecutar el contenido del ciclo *for* diez veces.

<code>for (i=0; i < 10; i++).....</code>	<i>Correcto</i>
<code>for (i=0; i <=10; i++).....</code>	<i>Incorrecto (ejecuta once veces)</i>

En [11] se presenta una revisión exhaustiva de las principales clasificaciones de errores en el ámbito de la programación y se profundiza en torno a nuevas maneras de entender la ejecución de un programa cuestionando las salidas del programa, también se realiza un análisis de algunos IDEs y lenguajes de programación usados en la actualidad y aporta una taxonomía de defectos comunes de la programación.

2.3. Sistemas expertos

Como una aplicación exitosa del área de Inteligencia Artificial (IA), surgieron los en la década de los 60 los denominados Sistemas Expertos; al principio, este tipo de sistemas se desarrolló como herramienta de investigación para tratar problemas

complejos en un dominio estrecho, por ejemplo, el diagnóstico médico de enfermedades en ese entonces, pero desde su introducción comercial a principios de los 80, incrementó su popularidad. Actualmente los sistemas expertos son usados en los negocios, ciencia, ingeniería, manufactura y muchos otros campos en los cuales existe un problema en un área o en un dominio bien definido. Feigenbaum, profesor de la Universidad de Stanford, y considerado como el padre de los sistemas expertos, presentó la siguiente definición para un sistema experto:

“...es un programa de cómputo inteligente que usa el conocimiento y procedimientos de inferencia para resolver problemas suficientemente complicados que requiere significativamente de un experto humano para su solución” [12].

De acuerdo con [13], en un sistema experto se pueden identificar tres componentes principales: base de conocimientos, motor de inferencia e interface de usuario. La figura 1 ilustra una posible arquitectura [14].

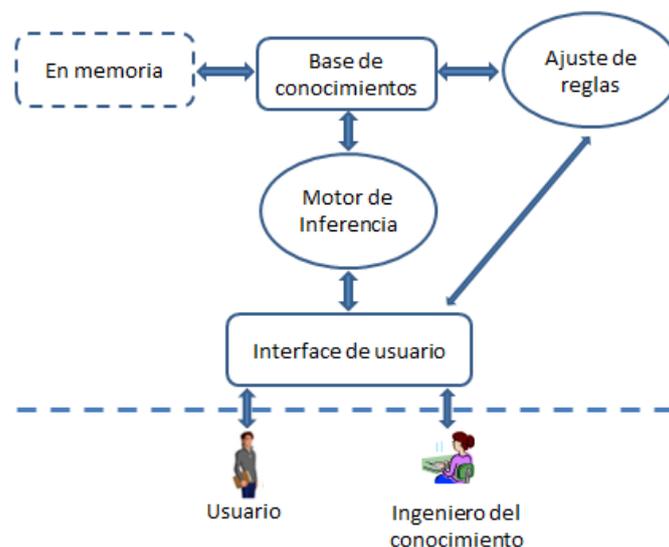


Fig. 1. Estructura de un sistema experto.

Base de conocimientos. Es la base de un sistema experto, contiene el conocimiento, datos del problema necesarios para entender, formular y suministrar propuestas de solución. Existen diferentes formas de representar el conocimiento de un sistema experto, incluyendo reglas de producción, redes semánticas y sentencias lógicas. Una base de conocimientos típica debe incluir dos elementos básicos:

- Hechos tal como la situación del problema y la teoría del área del problema.
- Heurísticas especiales o reglas que dirigen el uso del conocimiento para resolver los problemas específicos en un dominio particular.

Motor de Inferencia. Es el cerebro de un sistema experto, también se le conoce como estructura de control o el intérprete de reglas. Este componente es esencialmente un programa de computadora que proporciona una metodología para el razonamiento sobre la información en la base de conocimientos, en el área de memoria y para la formulación de conclusiones.

Interface de usuario. Este componente establece la comunicación orientada al problema entre el usuario y la computadora, dicha comunicación se realiza en un lenguaje natural. La interface de usuario en este tipo de sistemas, para su interacción, en su mayoría usa preguntas y respuestas. En la Figura 1 se visualizan dos tipos de usuarios:

- El usuario, quien utiliza el sistema experto para hacer consultas y ayudarlo a tomar decisiones.
- El ingeniero del conocimiento, quien se encarga del mantenimiento de la base de conocimientos, es la interface entre el humano experto y el sistema experto.

Como se observa en la figura 1, en [14] en dicha arquitectura se consideran dos elementos adicionales: lo que se mantiene en Memoria, y las reglas ajustadas durante la operación del sistema experto.

Memoria. El contenido de este elemento consiste de los hechos generados de acuerdo a un problema específico en una sesión de consulta, es decir los nuevos hechos que resultaron de un proceso de inferencia.

Ajuste de reglas. Se trata del ajustador de reglas, es decir la entrada de reglas especificadas por el ingeniero del conocimiento dentro de la base de conocimientos durante el desarrollo del sistema experto o mantenimiento de la base de conocimientos.

3. Metodología

Como parte de la investigación antes citada [1], se desarrolló un sistema experto basado en reglas, capaz de identificar los tipos de defectos comunes en la programación; dicho sistema experto se integró como un componente al Entorno Virtual Colaborativo (EVC) que fue diseñado exprefeso para la investigación, en particular para el proceso de experimentación. En términos generales, las actividades que se planificaron como parte de la metodología para la construcción del Entorno Virtual Colaborativo Inteligente (EVCI) son:

1. Diseño una Arquitectura para un EVCI.
2. Implementación del Entorno Virtual Colaborativo.
3. Formalización de las reglas que permitirán incorporar al EVC una Base de Conocimiento sobre los defectos comunes en programación.
4. Implementar el componente inteligente (Sistema Experto) al EVC.
5. Realizar estudios con experimentos controlados en ingeniería de software utilizando el EVCI.
6. Reporte de resultados.

4. Desarrollo

4.1. Arquitectura propuesta para el EVCI

Un Entorno Virtual Colaborativo Inteligente (EVCI) es un sistema computacional diseñado exprefeso como un espacio conceptual para que el usuario, en condiciones

espacio-temporales distintas, interacción con otros usuarios o con elementos del entorno para construir su aprendizaje; el componente inteligente es un elemento del entorno provisto de una base de conocimientos y de una estrategia pedagógica basada en la formulación de consultas por parte del aprendiz [1].

La Figura 2 ilustra mediante un diagrama la arquitectura que se diseñó y utilizó para la implementación del EVCI presentado en esta investigación.

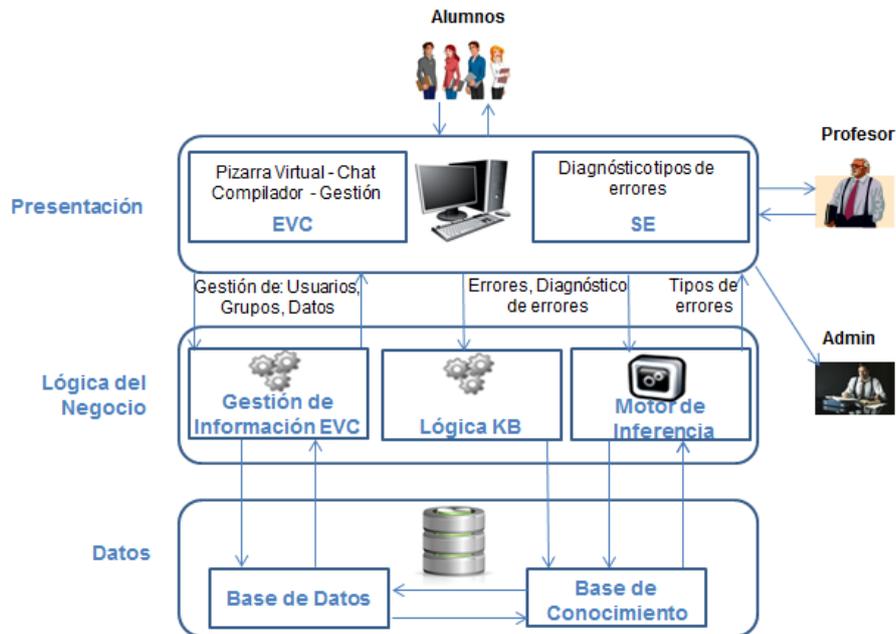


Fig. 2. Arquitectura del EVCI.

La arquitectura del EVCI consta de la integración de los ambientes de un entorno virtual colaborativo (EVC) y de un Sistema Experto (SE). Esta arquitectura se puede visualizar en una arquitectura por capas [15], donde los usuarios principales del sistema son los alumnos; los alumnos pueden visualizar en el EVC trozos de código de programas implementados bajo un paradigma de programación seleccionado (p.e. el paradigma estructurado) y deben encontrar los defectos del programa que se les presente, lo podrán realizar en grupos trabajo, consultando con el profesor o con un SE integrado al EVCI que permite identificar los tipos de defectos comunes de la programación.

4.2. Base de conocimiento

En esta sección, se establece la base de conocimiento sobre los tipos de defectos más comunes de la programación que se implementó en el sistema experto, componente inteligente del EVCI. En este trabajo se abordó el problema del aprendizaje y/o enseñanza de la programación mediante entornos virtuales, utilizando como parte del modelo del dominio, un modelo de perturbaciones o de defectos clásicos en el dominio del aprendizaje [16].

Para representar la base de conocimiento en esta investigación se utilizaron reglas de producción, estas reglas se generan relacionando los datos de la base de hechos. Cada regla está formada de una parte denominada premisa y de una parte denominada conclusión y tendrá la siguiente forma:

- Si premisa Entonces conclusión.

El sistema EVCI ha sido concebido para el entrenamiento de los aprendices en la identificación de defectos, en este sentido, debido a que la relación entre los tipos de defectos cometidos por los aprendices y el diagnóstico no tiene un comportamiento determinístico, se ha decidido considerar un modelo probabilístico para el sistema experto, es decir, su motor de inferencia se basa en probabilidades condicionales. En este esquema se consideran todas las evidencias, por lo que también se pueden modelar reglas de la forma:

- Si premisa1 And premisa2 Or premisa3 Entonces conclusión.

Los sistemas basados en reglas surgieron de sistemas prácticos e intuitivos para la inferencia lógica. Estos métodos fueron desarrollados a mediados de los setentas y han formado las bases de un gran número de sistemas expertos en medicina y en muchas otras aplicaciones [15].

Tabla 1. Atributos y valores de los defectos (errores) sistemáticos de la programación.

Diagnósticos de error (DxE)	Tipos de errores comunes o sistemáticos					
	Inicialización	Cómputo	Control	Datos	Interface	Cosmético
Diseño lógico		Caso imprevisto, comisión Error en el algoritmo, comisión	Terminación abrupta			Omisión
Sobrecarga de memoria	Subíndice fuera de limite, comisión	Liberación de memoria, omisión	Validación de memoria, omisión.		Terminación abrupta	
Léxico		Análisis inadecuado, comisión	Sintaxis ambigua, comisión			
Variable	Errores por tipos diferentes, comisión Inicialización inapropiada de una estructura de datos comisión	Resultados incorrectos, comisión		Resultados incorrectos, comisión		
Lenguaje		Interpretaciones erróneas en la semántica del lenguaje				

Una de las actividades primordiales para el desarrollo de un sistema experto, es la adquisición del conocimiento posteriormente codificado en la base de conocimientos,

una base de datos existente y apropiada dentro de un conjunto de reglas de producción. La fuente de este conocimiento son los defectos comunes de la programación, estudiado en el marco teórico.

Debido a la gran cantidad de lenguajes de programación, para poder realizar la experimentación con el EVCI, se decidió una versión reducida de la estructura identificada como fuente de conocimientos, en específico, en el lenguaje de programación C, los objetos son los seis tipos de defectos comunes de la programación propuestos en [17] y los atributos y valores de dichos objetos son los investigados en [11]. En la Tabla 1 se presenta una estructura utilizada para la construcción de la base de conocimientos.

Los defectos representan el aspecto más importante ya que son estos elementos los que en mayor medida permiten encontrar un tipo de defecto específico basado en la presencia o ausencia de ciertos indicadores.

Los diagnósticos de defecto son indicadores que también forman parte del proceso de obtención de tipos de defectos diferenciales. Los diagnósticos de defecto permiten a los profesores de programación o expertos, confirmar un tipo de defecto a través de los resultados reportados, es por eso que se han tomado en cuenta como un tipo de indicador.

En la siguiente lista se presenta la base de datos que almacena la base de conocimiento del sistema EVCI utilizado para el experimento, es modelado en forma de reglas:

Regla 0:	<i>Si</i> Error = Subíndice fuera de límite <i>And</i> Error = Comisión <i>And</i> DxE = Sobrecarga de memoria <i>Entonces</i> Tipo de error = Inicialización.
Regla 1:	<i>Si</i> Error = Caso imprevisto <i>And</i> Error = Comisión <i>And</i> DxE = Diseño lógico <i>Entonces</i> Tipo de error = Cómputo.
Regla 2:	<i>Si</i> Error = Error en el algoritmo <i>and</i> Error = Comisión <i>and</i> DxE = Diseño lógico <i>Entonces</i> Tipo de error = Cómputo.
Regla 3:	<i>Si</i> Error = Liberación de memoria <i>And</i> Error = Omisión <i>And</i> DxE = Sobrecarga de memoria <i>Entonces</i> Tipo de error = Cómputo.
Regla 4:	<i>Si</i> Error = Análisis inadecuado <i>And</i> Error = Comisión <i>And</i> DxE = Léxico <i>Entonces</i> Tipo de error = Cómputo.
Regla 5:	<i>Si</i> Error = Interpretaciones erróneas en la semántica del lenguaje <i>And</i> DxE = Lenguaje <i>Entonces</i> Tipo de error = Cómputo.

Regla 6:	<i>Si Error = Terminación abrupta And DxE = Diseño lógico Entonces Tipo de error = Control.</i>
Regla 7:	<i>Si Error = Validación de memoria And Error = Omisión And DxE = Sobrecarga de memoria Entonces Tipo de error = Control.</i>
Regla 8:	<i>Si Error = Resultados incorrectos And Error = Comisión And DxE = Variable Entonces Tipo de error = Datos.</i>
Regla 9:	<i>Si Error = Sintaxis ambigua And Error = Comisión And DxE = Léxico Entonces Tipo de error = Control.</i>
Regla 10:	<i>Si Error = Terminación abrupta And DxE = Sobrecarga de memoria Entonces Tipo de error = Interface.</i>
Regla 11:	<i>Si Error = Omisión And DxE = Diseño lógico Entonces Tipo de error = Cosmético.</i>

La razón de usar reglas de producción para la representación del conocimiento en esta investigación, es la facilidad para entender y agregar nuevas reglas a la base de conocimiento sin afectar las reglas que ya existan [13].

5. Resultados

Como resulta de esta investigación, se presenta uno de los componentes más significativos del Sistema EVCI, éste es el Sistema Experto. En la Figura 3 se visualiza una página de este componente, en esta página el profesor puede ver tres listas.

La primera corresponde a la lista de los tipos de defectos (errores). Cabe mencionar que este EVCI se ha utilizado para realizar estudios con experimentos controlados en Ingeniería de Software [18, 19], para los experimentos se escogió los seis tipos defectos comunes. La segunda lista corresponde a los defectos (errores) de cada una de los tipos de defectos y la tercera lista muestra los diagnósticos de error asociados a cada tipo de defecto.

Para ambos casos, tanto para la lista de tipos, la lista de defectos como para la de diagnósticos de defectos, en el menú inferior de la aplicación se muestran los botones correspondientes para agregar un registro nuevo a la lista correspondiente, también para cada registro es posible seleccionar y eliminar el dato correspondiente. Las opciones de agregar o eliminar datos solo son posibles con el rol de administrador o tutor, donde

este último es el experto que puede ser el profesor, un alumno solo puede realizar operaciones de consulta e inferencia.

Tipo de error	Fecha y hora de creación	Creador	Eliminar
Cosmético	2014-11-21 08:50:30	admin_1	
Inicialización	2014-10-31 00:30:12	admin_1	
Cómputo	2014-10-31 00:30:12	admin_1	
Control	2014-10-31 00:30:12	admin_1	
Datos	2014-10-31 00:30:12	admin_1	

Diagnóstico	Eliminar
Abstracción	

Diagnóstico	Eliminar
Abstracción	

Fig. 3. Página para la edición de base del conocimiento.

Al pulsar sobre el botón de errores sistemáticos sobre la misma página, se muestra un diálogo que se ilustra en la Figura 4.

En este diálogo el profesor puede seleccionar de la lista defectos (errores sistemáticos) el elemento que corresponda agregar, como parte del tipo de error que haya seleccionado previamente. Dado que esta lista es larga se ha puesto una caja de texto en donde los profesores escriben una palabra que el sistema utilizará como criterio de búsqueda y entonces reducirá la lista de defectos a solamente aquellos términos que coincidan con el criterio de búsqueda.

Para el caso de los diagnósticos de defectos el sistema funciona de la misma manera. Con el botón Diagnósticos de error el sistema permite a los profesores agregar nuevas pruebas diagnósticas a la lista y relacionarlas con el tipo de errores que se esté modelando. Al pulsar sobre el Diagnósticos de error aparecerá el diálogo que se muestra en la Figura 5 con un listado de todos los diagnósticos de defectos que se encuentran en la base de conocimientos original.

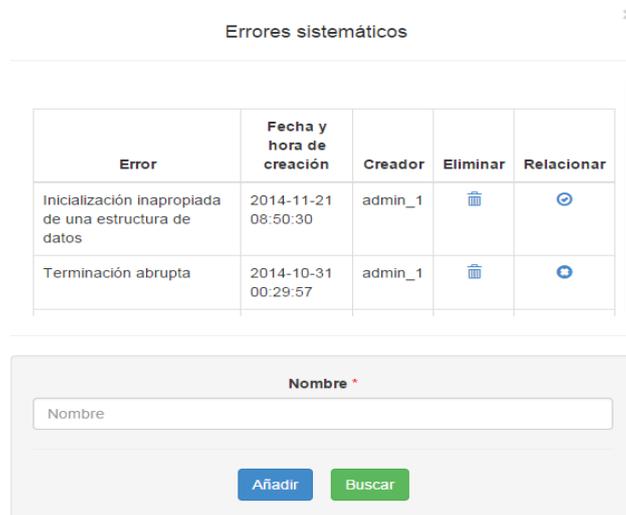


Fig. 4. Diálogo para agregar nuevos errores.

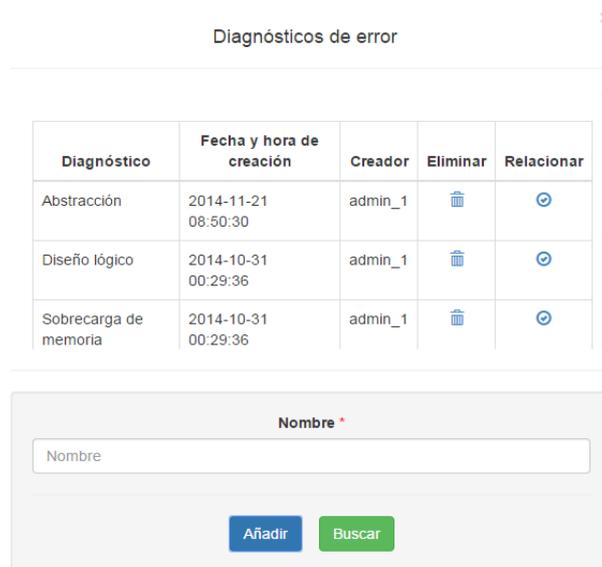


Fig. 5. Diálogo para agregar diagnósticos de defectos.

Dentro de este mismo módulo del Sistema EVCI, además de la opción que corresponde a la edición de la base de conocimiento, explicado anteriormente, también tiene una segunda opción que corresponde a la obtención de los tipos de defectos comunes de la programación, aquí tienen permisos tanto el alumno como el profesor y en su caso el administrador.

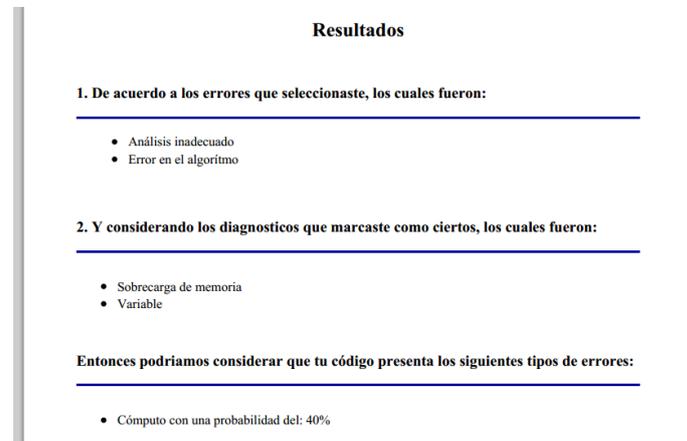


Fig. 6. Resultado con lista de los tipos de defectos diferenciales.

Después de haber agregado elementos a dichas listas, el contenido de estas listas se envía al motor de inferencia quien a su vez hace la búsqueda en la base de conocimiento para, finalmente, mostrar la lista de los tipos de defectos diferenciales que se generen como respuesta. Un ejemplo de esto se muestra en la Figura 6.

Como se puede observar, los resultados se imprimen con porcentajes debido a que en el método basado en reglas se consideró un sistema probabilístico y éste se calcula con base al número de atributos y diagnósticos de defectos que tiene un tipo de defecto.

6. Trabajos en curso y futuros

De acuerdo con las actividades descritas en la sección 3, el proceso de experimentación con el Entorno Virtual se ha comenzado a realizar en diversos escenarios, tal y como se ha reportado en [18], no obstante, el componente Inteligente (Base de Conocimientos) aún no ha sido validado; al momento de redactar este artículo el grupo de investigación se encuentra diseñando un escenario de aprendizaje que permita validar el uso del Sistema Experto.

Entre las conclusiones parciales de los experimentos realizados, se ha resuelto la conveniencia de experimentar con otros lenguajes de programación; en el diseño original la Base de Conocimientos se utilizó como referente al paradigma estructurado, en específico se utilizó el Lenguaje C; actualmente se está experimentando con otro lenguaje de programación, éste pertenece al paradigma de programación orientada a objetos, se utilizó Java, resultado parcial reportado en [19]. Otro trabajo interesante a futuro, sería adecuar el EVCI para que aprendiera en forma autónoma directamente de la experiencia del estudiante, para convertirla en una herramienta computacional mucho más flexible, sin dependencia del experto o del tutor.

Finalmente, una actividad obligada y que será realizada a mediano plazo, consiste en medir el desempeño del sistema experto del EVCI; lo cual puede ser realizado comparando el sistema propuesto con otros sistemas o con un estándar, y para ello se requerirá un conjunto de expertos.

Referencias

1. Ucán, J.: Aprendizaje de la Programación Asistido con Entornos Virtuales Colaborativos Inteligentes. Tesis Doctoral, Universidad del Sur (2015)
2. Pennington, N., Grabowski, B.: The tasks of programming. Hoc, J.-M., Green, T. R. G., Samurcay, R., Gilmore, D.J. (Eds), Psychology of programming, Academic Press, UK (1990)
3. Jablonowski, J.: Some Remarks on Teaching of Programming, Proceedings of International Conference on Computer Systems and Technologies (2004)
4. Kaasbøll, J.: Exploring didactic models for programming. Norsk Informatikk-Konferanse, Høgskolen I Agder (1998)
5. Rogalski, J., Samurcay, R.: Adquisition of programming knowledge and skills. Hoc, J.-M., Green, T. R. G., Samurcay, R., Gilmore, D.J. (Eds), Psychology of programming, Academic Press, UK (1990)
6. Vizcaino, A.: Enhancing Collaborative Learning Using a Simulated Agent. Tesis Doctoral, Universidad de Castilla-La Mancha, España (2002)
7. Guzdial, M.: Programming environments for novices. Fincher, S., Petre, M. (Eds.), Computer Science Education Research (pp. 127–154), Lisse, The Netherlands: Taylor & Francis, pp. 127–154 (1994)
8. Kelleher, C., Pausch, R.: Lowering the barriers to programming: a taxonomy of programming environments and languages for novice programmers. ACM Computing Surveys, Vol. 37, No. 2, pp. 88–137 (2005)
9. IEEE, IEEE Standard Glossary of Software Engineering Terminology. Std. 610.12-1990. Institute of Electrical and Electronics Engineers, NY, USA (1990)
10. Mayrhauser, A., Vans, A.: Program Understanding Behavior During Debugging of Large Scale Software, Empirical Studies of Programmers, 7th Workshop, Alexandria, VA pp. 157–179(1997)
11. Ko, A.: Asking and Answering Questions about the causes of Software Behavior. PhD Thesis, University of British Columbia (2008)
12. Giarratano, J., Riley, G.: Expert systems: principles and programming. Canada: Thomson Technology (2005)
13. Turban, E., Aronson, J.E., Liang, T.P., Sharda, R.: Decision Support System and Business Intelligence. Eighth Edition, Edit Pearson Education, Inc. (2012)
14. Ignizio, J.: Introduction to expert systems. McGraw-Hill (1991)
15. Hernández, G.: Análisis del uso de la inteligencia colaborativa como herramienta para la construcción de bases de conocimiento consensuadas en procesos de diagnóstico médico. Tesis Doctoral, Universidad Carlos III de Madrid (2013)
16. Holt, P., Dubs, S., Jones, M., Greer, J.: The state of Student Modelling. Student Modelling: The Key to Individualized Knowledge-Based Instruction, Springer-Verlag, pp. 3–39, (1994)
17. Basili, V., Perricone, B.: Software errors and complexity: an empirical investigation. Communications of the ACM, Vol. 27, No. 1, pp. 42–52 (1984)
18. Ucán, J., Gómez, O., Castillo, A., Aguilar, A.: Detección de defectos con y sin apoyo de un entorno virtual colaborativo inteligente en cursos introductorios de programación. Gómez, O.S., Arcos, G., Aguirre, L., Villa, E., Rosero, R.H. (eds), Ingeniería de Software e Ingeniería del Conocimiento. Jornadas Iberoamericanas. 11th (JIISIC'2015). Curran Associates, Inc., Riobamba, Elsevier, pp. 65–78 (2015)
19. Ucán, J., Aguilar, R., Aguilera, A., Díaz, J.: Analizando la efectividad del uso de un EVCI para asistir a estudiantes avanzados en la identificación de faltas en el código: un experimento controlado. Revista Latinoamericana de Ingeniería de Software, Vol. 4, No. 1 (2016)

Sistema tutor afectivo para el aprendizaje de las matemáticas usando técnicas de gamificación

Ramón Zatarain-Cabada, María Lucía Barrón-Estrada,
Jorge García-Lizárraga

Instituto Tecnológico de Culiacán, Sinaloa,
México

{rzatarain, lbarron, 86170331}@itculiacan.edu.mx

Resumen. De acuerdo a evaluaciones internacionales en los niveles básicos de educación, México se encuentra entre los países que presentan menor puntuación en lo que respecta al aprendizaje de las Matemáticas. Este artículo presenta el diseño de un Sistema Tutor Afectivo para el Aprendizaje de las Matemáticas (STAAM), el cual relaciona situaciones de aprendizaje con ejercicios matemáticos orientados al plan de estudios oficial vigente. El tutor incluye un reconocimiento bimodal de emociones, una técnica de gamificación para la motivación del estudiante y una retroalimentación en cada ejercicio de acuerdo al estilo de aprendizaje del usuario. La puntuación de cada ejercicio se determina con base a variables como el tiempo de respuesta, el número de intentos y el nivel del ejercicio. Se incorporó el estado emocional del estudiante para determinar el siguiente ejercicio. Los resultados obtenidos con estudiantes, mostraron un grado satisfactorio de aceptabilidad como también un avance significativo en ciertos aprendizajes esperados del plan de estudios.

Palabras clave: Computación afectiva, sistemas tutores inteligentes, gamificación, software educativo.

Affective Tutoring System for Learning of Mathematics Using Gamification Techniques

Abstract. According to international assessments in basic education levels, Mexico is among the countries with the lowest score in regard to the learning of mathematics. This paper presents the design of an Affective Tutoring System for Learning Mathematics (ATSLM), which relates learning situations with mathematical exercises oriented to the current official curriculum in Mexico. The tutor includes a bimodal emotion recognition, a gamification technique for student motivation and feedback for each exercise according to the user's learning style. The score for each exercise is determined based on variables such as response time, number of attempts and the exercise complexity level. Student's emotional state is incorporated to determine the next exercise. The results obtained with students, showed a satisfactory degree of acceptability as a significant advance in certain learning outcomes of the curriculum.

Keywords: Affective computing, intelligent tutoring systems, gamification, educational software.

1. Introducción

De acuerdo con los resultados de PISA 2012, el panorama en matemáticas de los jóvenes mexicanos es insatisfactorio ya que no alcanzan el nivel de competencias básico (nivel 2) en matemáticas (promedio OCDE: 23%). El alumno promedio en México obtiene 413 puntos en matemáticas. El puntaje promedio en la Organización para la Cooperación y el Desarrollo Económico (OCDE) es de 494, una diferencia que equivale a casi dos años de escolaridad [1].

Los Sistemas Tutores Afectivos (STA) son Sistemas Inteligentes que incorporan la habilidad para reconocer el estado afectivo de los estudiantes permitiendo al usuario interactuar con ejercicios que estimulen su estado emotivo [2].

Las técnicas de gamificación (gamification) se usan en la educación con el propósito de disminuir los abandonos, la falta de motivación y la falta de compromiso en las actividades de aprendizaje [3].

El uso de un STA que integre el reconocimiento emocional, técnicas de gamificación y estilos de aprendizaje [4], es una alternativa para el mejoramiento del aprendizaje de las matemáticas [5].

Este trabajo presenta un Sistema Tutor Afectivo para el Aprendizaje de las Matemáticas (STAAM) de primer grado de secundaria, usando ejercicios matemáticos de acuerdo al plan de estudios vigentes, con la incorporación de técnicas de gamificación que permiten motivar al estudiante, con técnicas de reconocimiento de emociones bimodal (texto y rostro), así como el uso del estilo de aprendizaje del usuario que permite dar una retroalimentación en cada ejercicio para favorecer su proceso de aprendizaje.

La implementación de STAAM para alumnos de Primer Grado de Secundaria permitirá:

- Despertar el interés del estudio de las matemáticas mediante el uso de un tutor afectivo implementando técnicas de gamificación.
- Fortalecer el aprendizaje de temas de difícil comprensión usando estrategias didácticas con retroalimentación orientadas a su estilo de aprendizaje.
- Integrar competencias matemáticas en alumnos de primer grado en su ingreso al nivel de secundaria.
- Evaluar el desenvolvimiento de los alumnos en cada uno de los aprendizajes esperados que contempla los 5 bloques del plan de estudios vigentes, identificando fortalezas y debilidades en cada uno de ellos.

De esta manera, la implementación de STAAM desarrolla una nueva forma de aprendizaje de las matemáticas de manera más motivante e innovadora.

Actualmente existe gran diversidad de herramientas de apoyo para el aprendizaje de las matemáticas que no contemplan la emoción de los usuarios al momento de utilizarlas. Diversos estudios demuestran que el estado emocional del usuario está estrechamente relacionado con procesos cognitivos como el aprendizaje [6], lo que puede favorecer o bloquear este proceso.

STAAM integra una plataforma para el reconocimiento de emociones en expresiones faciales y textuales denominado PREMOC.

Este artículo está dividido en 6 secciones. En la segunda sección se describen trabajos relacionados. En la tercera sección se presenta la arquitectura de STAAM así como el trabajo realizado en el desarrollo del reconocedor PREMOC que se utilizó en la investigación. En la cuarta sección se describe el funcionamiento del tutor. En la quinta sección se muestran resultados del software propuesto. Por último, en la sexta sección se describen las conclusiones del trabajo.

2. Trabajos relacionados

Existen diversos trabajos sobre el aprendizaje de las matemáticas que se enfocan solamente en la resolución de problemas matemáticos como MathTutor [7]. Esta herramienta es un Sistema Tutor Inteligente (STI) basado en la Web diseñado para ayudar a los estudiantes de 6to grado de primaria, 1ro y 2do de secundaria, para aprender Matemáticas realizando ejercicios¹.

C. González y A. Mora utilizan elementos de gamificación como componentes principales del sistema EMATIC (Mathematics Education through ICT) [8], el cual es un STI orientado a tablets y dispositivos móviles para la enseñanza de operaciones matemáticas básicas.

Por otra parte, E. Melis presenta un STI (ActiveMath) que proporciona recursos interactivos para mejorar y enriquecer la enseñanza y el aprendizaje de las matemáticas [9]. El STI es un sistema tutorial inteligente basado en la Web para las matemáticas que contiene actividades que pueden ser utilizadas tanto por los alumnos individuales, grupos pequeños o con toda la clase utilizando un proyector o una pizarra interactiva.

MathSpring (antes Wayang Outpost) [10], es un Sistema Tutor Inteligente diseñado para aprender junto con el estudiante. Usando multimedia interactiva, MathSpring ayuda a preparar a los estudiantes de secundaria y preparatoria para los exámenes estandarizados de matemáticas como el SAT, MCAS y CA-Star.

Animal Watch [11], es un STI que fue desarrollado con dos objetivos en mente: a) enseñar con eficacia aritmética y pre-álgebra para estudiantes de primaria y secundaria, b) aumentar la confianza, el valor y el gusto de las matemáticas de los jóvenes estudiantes.

S. D'Mello, A. Olney, C. Williams, y P. Hays desarrollaron un tutor que utiliza un rastreador ocular comercial para monitorear los patrones de la mirada de un estudiante e identificar cuando está aburrido, llamado Gaze Tutor [12]. Este STI tiene como objetivo promover la participación y el aprendizaje mediante la detección de forma dinámica y la respuesta al aburrimiento.

La principal aportación de STAAM es la integración en el aprendizaje de las Matemáticas de las técnicas de reconocimiento bimodal de emociones, con técnicas de gamificación y estilos de aprendizaje. Otra aportación importante de nuestra herramienta, es que fue creada tomando en cuenta el programa oficial de estudios de Matemáticas vigente en México, por lo que se adapta completamente a los ejercicios y problemas aplicados y usados por los profesores y estudiantes del nivel básico.

¹ <https://mathtutor.web.cmu.edu/>

3. Desarrollo del reconocedor bimodal de emociones

Las emociones se han considerado como un factor importante en la toma de decisiones [14], y el aprendizaje de las matemáticas no es la excepción. Por esta razón se consideró importante la incorporación del estado emocional del estudiante al momento de llevar a cabo los ejercicios matemáticos, lo cual permite dirigir el aprendizaje de acuerdo a su estado afectivo.

3.1. Arquitectura del STAAM

La estructura del Sistema Tutor Afectivo (Figura 1) presenta el funcionamiento general de STAAM desde la interfaz de usuario, los controladores (Afectivo, Gamificador y de Ejercicio), teniendo como aportación importante la incorporación del estado emocional para determinar el siguiente ejercicio a realizar por el estudiante, así como el controlador gamificador.

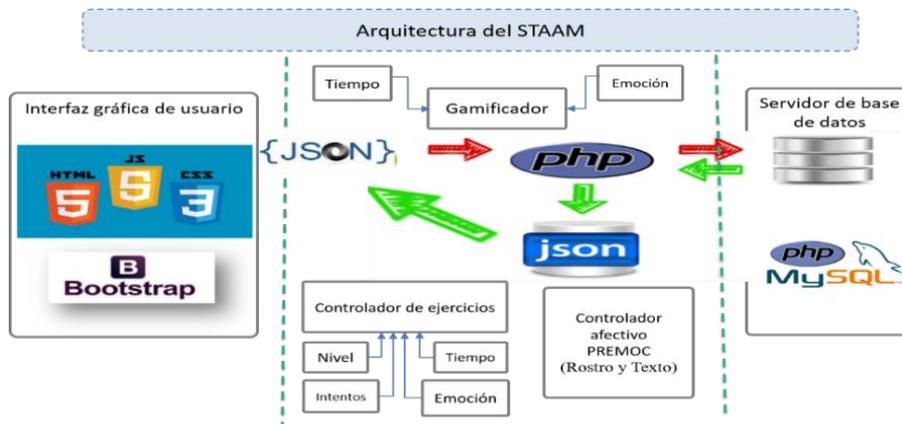


Fig. 1. Estructura del Tutor.

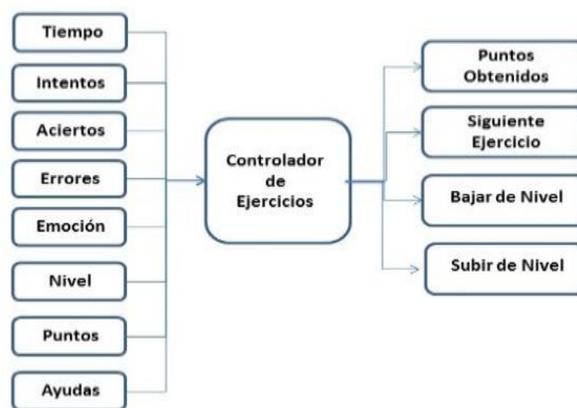


Fig. 2. Controlador de ejercicios.

STAAM integra la persistencia de la información en una base de datos en un servidor apache usando intercambio de información mediante objetos JSON.

3.2. Controlador de ejercicios

El controlador de ejercicios considera 8 variables importantes (Figura 2).

El controlador de ejercicios implementa 4 algoritmos (Figura 3), que nos permiten determinar: los puntos obtenidos por cada ejercicio, el siguiente ejercicio a realizar y el cambio de nivel de aprendizaje (bajar o subir de nivel).

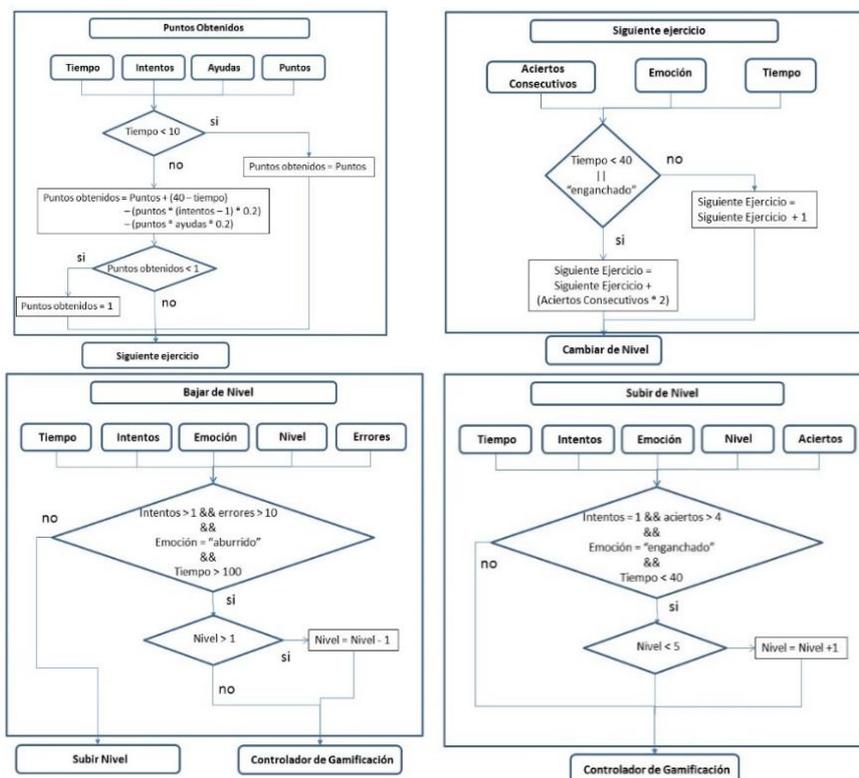


Fig.3. Algoritmos implementados en el controlador de ejercicios.

3.3. Controlador de gamificación

El controlador gamificador (Figura 4) se encarga de calcular los puntos extras o bonos que se identifican con estrellas personalizadas con una imagen, acorde a la emoción reconocida. Estos puntos extras o bonos se obtienen considerando el tiempo de respuesta (entre menos tiempo mayor puntaje), el número de aciertos realizados de forma consecutiva (entre mayores aciertos consecutivos se obtendrá mayor puntaje) y la emoción obtenida al momento de responder (cuando la emoción es “enganchado”).

Cuando la emoción detectada es “frustrado”, el tutor ofrece ayuda de retroalimentación del ejercicio propuesto, de acuerdo al estilo de aprendizaje del usuario y ejecuta ejercicios de menor nivel cognitivo si las últimas cuatro respuestas son erróneas y el tiempo de respuesta es mayor a 40 segundos.

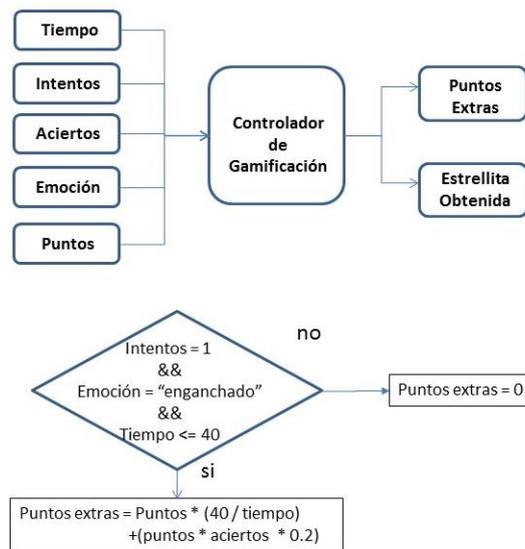


Fig. 4. Implementación del controlador de gamificación.

3.4. PREMOC

PREMOC es una plataforma en línea que ofrece reconocimiento de emociones a través de un servicio Web. La comunicación entre PREMOC y STAAM es a través de formato JSON, en donde el tutor envía sus datos de entrada por medio de un POST en este formato y recibe la emoción en modo “student”. En modo *student*, el PREMOC dará como respuesta el valor de una emoción tomando en cuenta el ámbito de aprendizaje (contexto educativo). Estas emociones son enganchado, aburrido, frustrado y neutral.

PREMOC tiene definidos cada uno de los reconocedores de forma autónoma, los cuáles reciben datos como: la imagen del rostro, voz, texto y/o información cerebral, este lo procesa, y obtienen la emoción. Los reconocedores fueron creados por separado en el laboratorio de la Maestría en Ciencias de la Computación del Instituto Tecnológico de Culiacán.

3.5. Reconocimiento de emociones en texto

Para el reconocimiento de emociones en texto, PREMOC utiliza un algoritmo semántico denominado ASEM. Este algoritmo está basado en palabras claves llamadas “keywords”, pues contiene una colección de palabras previamente etiquetadas a una emoción.

Para llevar a cabo el algoritmo ASEM se cuenta con 5 diferentes corpus, con diferentes funciones cada uno de ellos: stop words, palabras impropias, corpus emocional, new words y corpus semántico.

- Stop words. Es una colección de palabras que no influyen en la emoción de la oración, como artículos, preposiciones, entre otros. Las palabras que están aquí son eliminadas de la oración a evaluar.
- Palabras impropias. Es una colección de, como su nombre lo dice, palabras impropias, que están calificadas a una emoción y su factor de probabilidad afectiva a dicha emoción (FPA) donde $0 < FPA < 1$.
- Corpus emocional. Es una colección de palabras, que están calificadas a una emoción y su FPA a dicha emoción, donde $0 < PFA < 1$.
- New words. Aquí se coleccionan las palabras nuevas que no se tienen en el corpus emocional. Esto es para el mantenimiento del algoritmo ya que las nuevas palabras encontradas aquí serán calificadas cada cierto tiempo por un experto. El experto determinará a que emoción pertenece esa palabra y su FPA.
- Corpus semántico. Es una colección de palabras que potencializa o inhibe el valor FPA semánticamente. Por ejemplo, la palabra muy potencializa el valor FPA de la siguiente palabra.

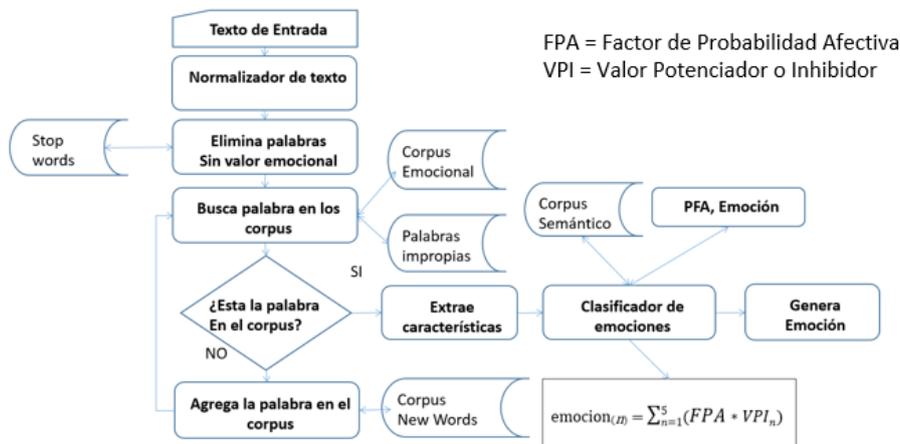


Fig. 5. Algoritmo ASEM.

Como se muestra en la imagen (Figura 5) el algoritmo ASEM funciona de la siguiente manera:

- Se recibe un texto de entrada.
- Se normaliza el texto quitando palabras que no agregan valor emocional usando el corpus *Stop Words*.
- Se busca cada palabra en el corpus emocional para obtener su emoción y su FPA. Si no está, se busca en el corpus *palabras impropias*. Si no está, se agrega al corpus *new words* para que sea más adelante clasificada por un experto. En la evaluación presente, dicha palabra es ignorada para la clasificación de la emoción en el texto.

- Se buscan palabras en el *corpus semántico* que puedan potencializar o inhibir los valores FPA ya obtenidos.
- Se calcula el valor de cada una de las 4 emociones del modo *student* (enganchado, frustrado, sorprendido y neutral) con la fórmula:

$$\text{emocion}_{(n)} = \sum_{n=1}^5 (FPA * VPI_n).$$

Se selecciona la emoción con el mayor valor obtenido y esa emoción se regresa al usuario. La sintaxis en JSON para indicar que lo que se envía es un texto es la siguiente:

txt:”texto a evaluar”.

3.6. Reconocimiento de emociones en rostros

El reconocedor de expresiones faciales en PREMOC consiste en clasificar las imágenes recibidas y regresar una de las 4 emociones en modo *student*.

La fase de ejecución en línea funciona de la siguiente manera como muestra en la Figura 6.

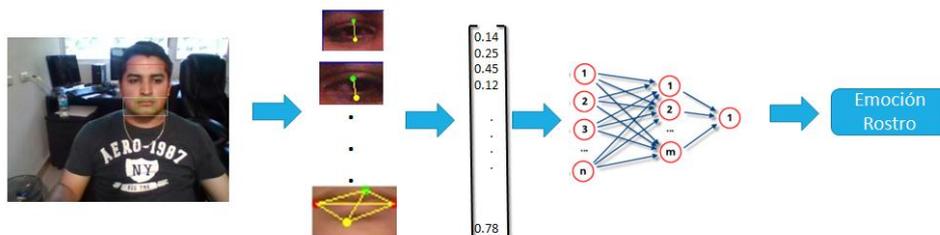


Fig. 6. Algoritmo del reconocedor de emociones en rostros.

- Recibe una imagen.
- Detecta el rostro en la imagen.
- Repite los siguientes pasos para boca, ojo izquierdo, ojo derecho, ceja izquierda, ceja derecha, lo llamaremos objetivo:
 - Ubicar dentro del rostro posible zona de ubicación del objetivo, se llamará región de interés.
 - Dentro de la región de interés aplicar un método de OpenCV para buscar en concreto el objetivo que estamos buscando.
 - Ya con el objetivo encontrado, se le aplican filtros de OpenCV [30] como se muestra en la figura 3 para determinar los puntos clave en el rostro.
- Se crea un vector de características de 10 elementos, donde cada uno es una distancia euclidiana entre diferentes puntos clave ubicados en el rostro.
- Se ingresa el vector de características a la red neuronal ya entrenada para que lo clasifique a una de las emociones.
- Regresa la emoción obtenida.

La sintaxis en JSON para indicar que lo que se envía es una imagen es la siguiente:

img:”Base64_imagen”.

El reconocedor de emociones en rostros la convierte de nuevo en una imagen para procesarla en modo “student”.

4. Implementación de STAAM

STAAM contiene diversos ejercicios divididos en 5 bloques, donde cada bloque puede contener hasta 12 secuencias didácticas. Cada secuencia didáctica puede tener hasta 12 temas principales y cada tema contiene un banco de reactivos de 1 a 999 ejercicios. (Figura 7).

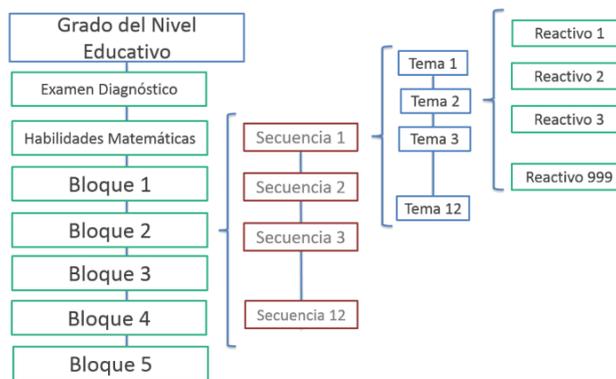


Fig. 7. Características del STAAM.

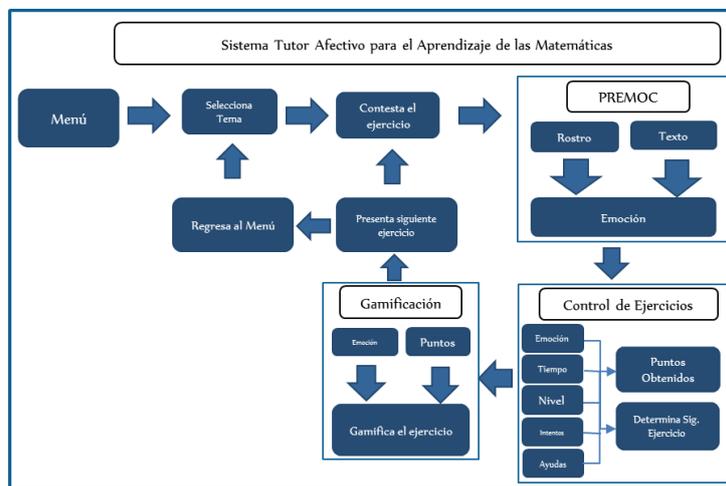


Fig. 8. Vista de controladores.

El estudiante resolverá los problemas que el STI le proporcione de acuerdo al determinado por el controlador de ejercicios. Cabe destacar que los contenidos que se abordan en cada uno de los ejercicios corresponden al plan de estudios vigentes. A

continuación, se muestra la arquitectura de los controladores principales de STAAM (Figura 8).

STAAM presenta el menú principal donde el estudiante selecciona el tema y la herramienta, usando el banco de ejercicios, le presenta el problema a realizar. Al momento de responder se toma una foto del rostro del estudiante para identificar su estado emocional, mediante la plataforma PREMOC. Ésta regresa la emoción del estudiante, la cual puede ser: enganchado, frustrado, sorprendido o neutral.

El controlador de ejercicios determina los puntos obtenidos por el estudiante, el siguiente ejercicio y el cambio de nivel de aprendizaje.

El controlador de gamificación considera la emoción y los puntos obtenidos para determinar el tipo de estrella ganado por el usuario, así mismo muestra el avance del ejercicio, los puntos globales obtenidos y la calificación obtenida por cada tema de acuerdo a la siguiente fórmula. (Figura 9)

Calificación =	Puntos obtenidos por cada tema	Nivel 1 = 10 pts Nivel 2 = 20 pts Nivel 3 = 40 pts Nivel 4 = 70 pts Nivel 5 = 100 pts
	----- Puntos del Nivel	

Fig. 9. Fórmula para la Calificación.

Por último, se muestra el siguiente ejercicio a evaluar y el proceso se repite.

5. Pruebas y experimentos

Actualmente se están haciendo experimentos con grupos de 1er año de secundaria (grupo A de experimentación y grupo B de control) obteniéndose resultados satisfactorios donde el uso de la herramienta ha despertado un gran interés por parte de los profesores y de los estudiantes.



Fig. 10. Grupo de experimentación 1.

En una prueba inicial (Fig. 10), participaron 19 estudiantes quienes terminaron los 20 ejercicios propuestos en un tiempo mínimo y con pocos errores, lo que permitió el aprendizaje de los ejercicios motivando a los estudiantes en la utilización del STAAM.



Fig. 11. Grupo de experimentación 2.

En una segunda intervención (Fig. 11), se invitó a madres de familia para que observarán la realización de ejercicios usando el tutor. Aquí participaron un grupo de 12 estudiantes, 8 mujeres y 4 hombres. Se trabajó con 40 ejercicios y se dispuso de un módulo de 45 min.

En la Tabla 1 se muestran los resultados obtenidos por 9 estudiantes escogidos del # lista (se escogieron los # de lista par).

Tabla 1. Resultados obtenidos.

Estudiante	Genero	Tiempo prom. De respuesta	# de intentos totales	Emoción Predominante	Nivel del Ejercicio
E01	M	43 seg	23	Neutral	Fácil
E02	M	48 seg	25	Frustrado	Fácil
E03	F	47 seg	24	Neutral	Fácil
E04	M	33 seg	21	Enganchado	Fácil
E05	F	38 seg	22	Neutral	Fácil
E06	M	43 seg	22	Enganchado	Fácil
E07	F	41 seg	22	Enganchado	Fácil
E08	F	45 seg	24	Frustrado	Fácil
E09	M	39 seg	22	Enganchado	Fácil

El tiempo de respuesta promedio de los estudiantes se eleva considerablemente cuando no se les permite el uso de calculadora llegando a ser por arriba de los 120 segundos en más del 80% de la población ocasionado por la dificultad de poder realizar las operaciones básicas de forma mental, mejorándose significativamente esta variable cuando si se utiliza la calculadora.

Para acceder al tutor es necesario autenticarse, si el usuario es nuevo, se procede a registrarse, de lo contrario debe ingresar su usuario y contraseña.

Al entrar el estudiante, debe seleccionar el bloque con el que se trabajará, así como el tema y subtema.

Una vez escogido el subtema, se procede a realizar los ejercicios correspondientes mostrándose en su caso, diferentes estrellas que identifican los ejercicios resueltos correctamente, los puntos y la emoción obtenida en el momento de la respuesta. (Figura 12).



Fig. 12. Pantalla de ejercicios.

Tabla 2. Estrellas indicadoras de los puntos ganados de acuerdo al tiempo y estado emocional.

Estrella ganada							
Emoción	frustrado	neutral	neutral	enganchado	enganchado	enganchado	sorprendido
Tiempo	Mucho	Mucho	Poco, Regular	Mucho	Regular	Poco	Poco, Regular
Puntos Extras	0	0	0	Ptos extras = pto ejerc + tiempo ejerc. - tiempo respuesta			

*Nota: Los puntos se obtienen solamente cuando el ejercicio se respondió correctamente.

Cada ejercicio realizado se identifica con icono representativo mediante una estrella la cual ésta relacionada a un estado emocional y el tiempo que tardó en contestar correctamente dicho ejercicio (Tabla 2).

6. Conclusiones

Se puede concluir que la implementación del tutor vendrá a favorecer el aprendizaje de las matemáticas ya que desafía al estudiante en cada uno de los temas propuestos manteniéndolos motivados en cada momento lo que permite su integración al tutor y desarrollar sus habilidades matemáticas.

El STAAM se encuentra en experimentación con un grupo A experimental de estudiantes de 36 alumnos, los cuales se dividen a su vez en tres sub-grupos de 12 alumnos.

Existe un grupo B con 33 estudiantes que se considerará como grupo de control a los cuales se les impartirá clases de forma tradicional usando las mismas secuencias didácticas y se les proporcionarán los mismos problemas a resolver. Esto permitirá hacer comparativos que permitan detectar como la herramienta proporciona un mayor nivel de efectividad con respecto al aprendizaje de las matemáticas.

Referencias.

1. OCDE. http://www.oecd.org/centrodemexico/medios/Mexico%20Country%20Note_SPANISH_final%20GR1_EGcomments_02_12_2013%20final.pdf. Bajado el 26/ mayo/ 2015 (2012)
2. Gonzalo Villareal, G.: Agentes inteligentes en educación. EDUTEC: Revista electrónica de tecnología educativa, Vol. 16, No. 4 (2003)
3. González González, C.S., Mora Carreño, A.: Técnicas de gamificación aplicadas en la docencia de Ingeniería Informática. Departamento de Ingeniería Informática, Universidad de La Laguna.
4. Barrón-Estrada, M.L., Zatarain-Cabada, R. Margarita Aranda-Ortega, H.E. Gómez-Pérez, José Mejía-Arredondo. Un tutor inteligente, afectivo y configurable para el aprendizaje de números naturales de 3er grado. *Research in Computing Science*, Vol. 77, pp. 45–54 (2014)
5. Clara Inés Peña, Jose-L Marzo, Josep Lluís de la Rosa, Ramón Fabregat: Un sistema de tutoría inteligente adaptativo considerando estilos de aprendizaje. *Revista UIS ingenierías*, 1(2) (2012)
6. Belavkin, Roman V.: The role of emotion in problem solving. *Proceedings of the AISB'01 Symposium on emotion, cognition and affective computing*, Heslington, York, England, pp. 49–57 (2001)
7. Alevin, V., McLaren, B. M., Sewall, J.: Scaling up programming by demonstration for intelligent tutoring systems development: An open-access web site for middle school mathematics learning. (2009)
8. González, C., Mora, A., Toledo, P.: Gamification in intelligent tutoring systems. *Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality*, pp. 221–225, ACM (2014)
9. Melis, E., Siekmann, J.: Activemath: An intelligent tutoring system for mathematics. *Artificial Intelligence and Soft Computing-ICAISC 2004*, pp. 91–101, Springer Berlin Heidelberg (2004)

10. V. Aleven, B. M. McLaren, J. Sewall: Scaling up programming by demonstration for intelligent tutoring systems development: An open-access web site for middle school mathematics learning. *IEEE Transactions on Learning Technologies*, Vol. 2, Issue 2 (2009)
11. Arroyo, I.: *Animalwatch: an arithmetic ITS for elementary and middle school students*. Proceedings of the 5th International Conference on Intelligent Tutoring Systems, Montreal, Canada (2000)
12. D'Mello, S., Olney, A., Williams, C., Hays, P.: Gaze tutor: A gaze-reactive intelligent tutoring system. *International Journal of human-computer studies*, 70(5), 377–398 (2012)
13. D. M. Hart, I. Arroyo, J. E. Beck, B. P. Woolf, C. R. Beal: *WhaleWatch : An intelligent multimedia math tutor*. (1999)
14. Celso M. de Melo, Jonathan Gratch, Peter J. Carnevale: *Humans versus Computers: Impact of emotion expressions on people's decision making*. (2015)

PREMOC: Plataforma de reconocimiento multimodal de emociones

Ramón Zatarain-Cabada, María Lucia Barrón-Estrada, Gilberto Muñoz-Sandoval

Instituto Tecnológico de Culiacán, Culiacán, Sinaloa,
México

{rzaratain, lbarron, m14170101}@itculiacan.edu.mx

Resumen. En años recientes la computación afectiva ha venido a mejorar la interacción humano-computadora, pues ayuda a la computadora a conocer el estado afectivo del usuario para mejorar la toma de decisiones. Este artículo presenta los avances en el proyecto PREMOC, una plataforma que brinda un servicio web para el reconocimiento de emociones en texto, imágenes de rostros, sonidos de voz y señales EEG de manera mono-modal y multimodal. PREMOC ayuda a los desarrolladores a integrar el reconocimiento de afecto a sus aplicaciones o sistemas de software. Cada uno de los reconocedores se implementó aplicando diferentes técnicas tanto para extraer características como para clasificar emociones; además para el reconocimiento multimodal se integraron las emociones mediante un sistema difuso. Esta plataforma ya está siendo utilizada por diferentes proyectos en el laboratorio de la Maestría en Ciencias de la Computación del Instituto Tecnológico de Culiacán.

Palabras claves: Computación afectiva, inteligencia artificial, reconocimiento de emociones.

PREMOC: Multimodal Emotion Recognition Platform

Abstract. In recent years, affective computing has been improving human-computer interaction, because it helps computers to know the emotional state of the user to improve the decision-making process. This paper presents the advance in the PREMOC project, a platform that provides a web service for emotion recognition in text, facial images, speech and EEG signals in mono and multimodal mode. PREMOC assists developers to incorporate emotion recognition on their applications. Each recognizer was implemented with different techniques for feature extraction and emotion classification; for multimodal recognition, emotions were integrated through a fuzzy system. This platform is being used on different projects in the Computer Science Laboratory of Instituto Tecnológico de Culiacán.

Keywords: Affective computing, artificial intelligence, emotion recognition.

1. Introducción

Las computadoras están adquiriendo desde hace varios años la capacidad de detectar la emoción de sus usuarios con el uso de diferentes sensores y dispositivos. Esto facilita la interacción entre el usuario y la computadora para crear una línea de comunicación de dos canales, con el cual la computadora obtiene información adicional del usuario. A esta nueva área del conocimiento se le nombra comúnmente como Computación Afectiva [1]. Hoy en día se han desarrollado una gran cantidad de proyectos y aplicaciones en diferentes ámbitos como la mercadotecnia, la medicina, la educación, entre otras.

Uno de los dispositivos más usados para detectar emociones para la computación afectiva es la cámara, pues según Mehrabian [2] los gestos faciales transmiten el 58% de la comunicación humana. Otra razón por la cual es la más usada es la existencia de características universales en los rostros que reflejan ciertas emociones sin importar la cultura facial de la persona; estas emociones son conocidas como las emociones de Ekman [3].

Otro dispositivo utilizado en la computación afectiva es el micrófono, pues la evaluación de la voz humana también ha sido utilizada para definir emociones como por ejemplo en Kun Han [4], donde al igual que muchos trabajos relacionados con el sonido, dividen el audio en pequeños segmentos y con ellos entrenan una red neuronal. En este método existen dos variables muy importantes, la intensidad y la valencia.

También se puede detectar emociones a partir del texto o documentos, ya que con el uso del internet, redes sociales y blogs, la cantidad enorme de información en estos medios abre una ventana muy grande para encontrar nuevos elementos dentro de estos datos, como es en este caso las emociones o afecto. Existen distintos métodos en diferentes idiomas de como reconocer emociones en el texto. Uno de los métodos más utilizados es el de la técnica keywords [5], que consiste en tener una base de datos previamente clasificada que contiene palabras etiquetadas con una emoción y un grado de pertenencia.

Un dispositivo más que podemos encontrar en la literatura es la diadema que emite señales electroencefalográficas (EEG). Este tipo de dispositivos puede ser muy efectivo para encontrar señales en tiempo real como se hace en Liu [6] donde se identificó las regiones del cerebro más importantes para interpretar emociones en un usuario.

Cada uno de estos dispositivos alcanza una tasa de asertividad de la emoción dependiendo de su implementación. Hay trabajos que combinan dos o más dispositivos para mejorar esa tasa de asertividad o para considerar diferentes aspectos del usuario antes de determinar su emoción. A la técnica donde se usan dos o más dispositivos o sensores para reconocer emociones, se le nombra reconocedor multimodal de emociones o afecto.

PREMOC ofrece cuatro reconocedores de emociones en forma de servicio web, es decir, cualquier aplicación cliente puede consultar al servicio web ofrecido por PREMOC y obtener una emoción de una imagen de un rostro, de un texto, de un sonido de voz o de un archivo de señales electroencefalográficas. Una de las ventajas de PREMOC es su modo multimodal, donde el usuario puede enviar en la misma consulta, dos o más diferentes tipos de archivos para obtener una emoción o cada una por separado. Otra ventaja es su gran flexibilidad y facilidad para ser usado, lo cual se da por medio de un servicio Web. Finalmente, otra ventaja para usar esta herramienta es

la identificación de emociones en texto y voz para el lenguaje Español, lo cual no existe en otros reconocedores multimodales de emociones, en el estudio que llevamos a cabo.

Este artículo está organizado de la siguiente manera: en el capítulo 2 se describen trabajos relacionados a PREMOC, tanto reconocedores individuales, como multimodales y algunos ejemplos que ofrecen el servicio en línea. En el capítulo 3 se describe cada uno de los reconocedores de manera individual, así como también se habla de los métodos que tiene PREMOC para la integración multimodal de emociones y la estructura de la plataforma. En el capítulo 4 se mencionan los experimentos y en el capítulo 5 las conclusiones y trabajo futuro.

2. Trabajos relacionados

Existen actualmente muchos trabajos relacionados a la obtención de emociones tanto monomodales como multimodales, y muchos de ellos trabajan con diferentes métodos de extracción y clasificación de características.

Entre los que detectan emociones por medio del texto, Seol [5] utiliza el algoritmo keywords para extraer características de las palabras y una red neuronal artificial basada en conocimiento (KBANN por sus siglas en inglés) para clasificarlas y obtener una emoción. Así como Shaheen [7] extrae las características del texto mediante reglas de reconocimiento de emociones y utiliza k-vecinos cercanos (KNN por sus siglas en inglés) para la clasificación de la emoción. También el método Random Forest fue usado por Zheng [8] para clasificación de emociones en el texto.

Por medio del rostro, Khandait [9] y Gangwar [10] utilizan una red neuronal para la clasificación de las características y obtención de emociones. Para la extracción de características utilizan SUSAN edge detection y Hue-Saturation-Value respectivamente. Mientras que De [11] usa distancia euclidiana para obtener la emoción y 2D Discrete Cosine Transform para la extracción de características.

En los trabajos relacionados a las emociones por medio de voz, Kostoulas [12] utiliza el algoritmo C4.5 de la librería Weka para la clasificación de la emoción. Vogt [13] con EmoVoice utiliza una máquina de soporte vectorial (SVM por sus siglas en inglés) por la rapidez para hacerlo en tiempo real, ya que EmoVoice es una aplicación en tiempo real. Caballero-Morales [14] utiliza Modelos Ocultos de Markov para la clasificación de la emoción (HMM por sus siglas en inglés) en su trabajo para el idioma español. Los 3 trabajos presentados utilizan Mel-Frequency Cepstral Coefficient (MFCC) como método de extracción de características.

En los trabajos donde se obtiene la emoción de una persona por medio de señales EEG los métodos son más variados. Por ejemplo, Liu [6] utiliza directamente los datos ofrecidos por Emotive Software Development Kit para la extracción de características y hace la clasificación de las emociones con un Modelo de Dimensión Fractal. Nie [15] obtiene las características EEG por medio de la Transformada Rápida de Fourier (FFT por sus siglas en inglés) y clasifica las emociones con SVM. Lee [16] utiliza NeuroScan para la extracción de características en las señales y las clasifica en emociones con Análisis Discriminante Cuadrático.

Hay trabajos bimodales como Alonso-Martin [17] que creó un sistema para la detección de emociones durante la interacción Humano-Robot, para que los robots sean más sociales, y utiliza como parámetros el rostro y la voz humana. También Busso [18]

creó un analizador bimodal de voz y rostros para demostrar que al usar ambos, la asertividad de la emoción dada es mejor que en modo individual. El autor utilizó SVM para clasificar emociones en el rostro y Praat para la voz. Chuang [19] trabajó con voz y texto, y usó SVM para la clasificación de emociones en la voz y keywords para el texto.

Existen también sistemas multimodales como el creado por D'Mello[20] para añadirle afecto al famoso sistema tutor inteligente AutoTutor donde además de utilizar voz y rostro, también tomaron en cuenta el lenguaje corporal. Caridakis [21] también utiliza estos 3 parámetros para la obtención multimodal de emociones y alcanza un 88,2% de asertividad, y clasifica 8 diferentes emociones con redes bayesianas. También hay investigación sobre aprovechar todo lo que se utiliza mientras se trabaja. Por ejemplo Paredes [22] trabaja en su investigación con mouse y teclado además de la cámara para la emoción a través del rostro.

En el trabajo realizado por Wagner [23] se introduce el nombre de Online Emotion Recognition System (OER) para referirse a los sistemas que ofrecen reconocimiento de emociones en línea. Wagner presenta Smart Sensor Integration que es un framework que ofrece reconocimiento de emociones de manera multimodal para voz, rostros y movimiento en línea. Otro framework multimodal es ABE [24] que consiste en una federación de agentes en línea donde cada agente se encarga de un dispositivo de entrada. ABE trabaja usando señales del cerebro (EEG), rastreo de ojos, expresiones faciales, conductividad en la piel, presión arterial y postura corporal.

3. PREMOC

PREMOC es una plataforma en línea que ofrece reconocimiento de emociones en los dispositivos anteriormente mencionados, lo cual hace a través de un servicio web para que los desarrolladores agreguen reconocimiento de emociones a sus aplicaciones. La comunicación entre PREMOC y los usuarios es a través de formato JSON, en donde el usuario envía sus datos de entrada por medio de un POST en formato JSON, y recibe la emoción como respuesta.

1.1. Reconocedores de emociones

PREMOC tiene definidos cada uno de los reconocedores de forma autónoma, los cuáles reciben el archivo con datos de la imagen, voz, texto y/o información cerebral, lo procesan, y obtienen la emoción que regresan. Los reconocedores fueron creados por separado en el laboratorio de la Maestría en Ciencias de la Computación del Instituto Tecnológico de Culiacán, y algunos ya fueron utilizados en una aplicación [26, 27].

1.1.1 Reconocedor de emociones en texto

Para el reconocimiento de emociones en texto, PREMOC utiliza un algoritmo semántico denominado ASEM. ASEM es un algoritmo basado en keywords, pues contiene una colección de palabras previamente etiquetadas a una emoción.

El algoritmo ASEM cuenta con 5 diferentes corpus, con diferentes funciones cada uno de ellos: stop words, palabras impropias, corpus emocional, new words y corpus semántico.

Stop words. Es una colección de palabras que no influyen en la emoción de la oración, como artículos, preposiciones, entre otros. Las palabras que están aquí son eliminadas de la oración a evaluar.

Palabras impropias. Es una colección de, como su nombre lo dice, palabras impropias, que están calificadas a una emoción y su factor de probabilidad afectiva a dicha emoción (FPA) donde $0 < FPA < 1$.

Corpus emocional. Es una colección de palabras, que están calificadas a una emoción y su FPA a dicha emoción, donde $0 < PFA < 1$.

New words. Aquí se coleccionan las palabras nuevas que no se tienen en el corpus emocional. Esto es para el mantenimiento del algoritmo ya que las nuevas palabras encontradas aquí serán calificadas cada cierto tiempo por un experto. El experto determinará a que emoción pertenece esa palabra y su FPA.

Corpus semántico. Es una colección de palabras que potencializa o inhibe el valor FPA semánticamente. Por ejemplo, la palabra muy potencializa el valor FPA de la siguiente palabra.

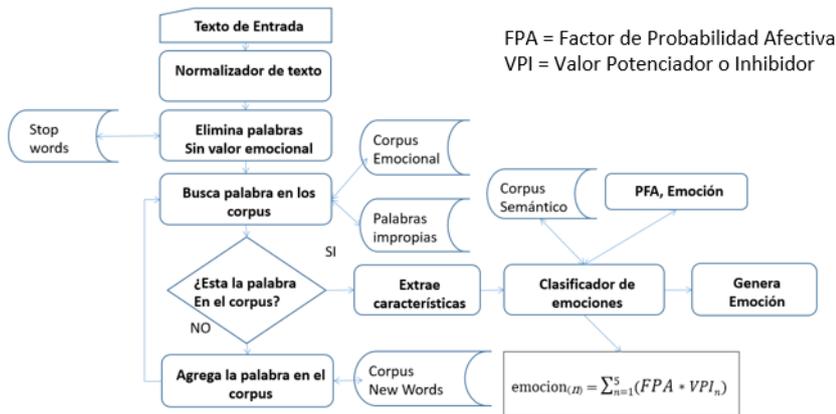


Fig. 1. Algoritmo ASEM.

Como se muestra en la Figura 1, el algoritmo ASEM funciona de la siguiente manera:

- a) Se recibe un texto de entrada.
- b) Se normaliza el texto quitando palabras que no agregan valor emocional usando el corpus Stop Words.
- c) Se busca cada palabra en el corpus emocional para obtener su emoción y su FPA. Si no está, se busca en el corpus palabras impropias. Si no está, se agrega al corpus new words para que sea más adelante clasificada por un experto. En la evaluación presente, dicha palabra es ignorada para la clasificación de la emoción en el texto.
- d) Se buscan palabras en el corpus semántico que puedan potencializar o inhibir los valores FPA ya obtenidos.

- e) Se calcula el valor de cada una de las 5 emociones (feliz, sorprendido, neutral, triste, enojado) en el texto con la fórmula:

$$\text{emocion}_{(n)} = \sum_{n=1}^5 (FPA * VPI_n).$$

- f) Se selecciona la emoción con el mayor valor obtenido y esa emoción se regresa al usuario.

1.1.2. Reconocedor de emociones en rostro

El reconocedor de expresiones faciales en PREMOC consiste de dos fases: entrenamiento de una red neuronal y uso del reconocedor.

La fase de entrenamiento consistió en entrenar la red neuronal encargada de la clasificación de rostros. Para esto se tomaron como datos de entrenamiento las imágenes del corpus RaFD [27]. Del corpus se tomaron 955 imágenes, 191 de cada una de las 5 emociones con las que se va a trabajar que son feliz, sorpresa, neutral, enojado y triste. De cada una de las imágenes se extrajo un vector de 10 características, donde cada una es una distancia euclidiana en diferentes puntos del rostro. Con esto se entrenó una red neuronal usando la librería Weka [28] para Java.

La fase de ejecución en línea funciona de la siguiente manera y como muestra la Figura 2.

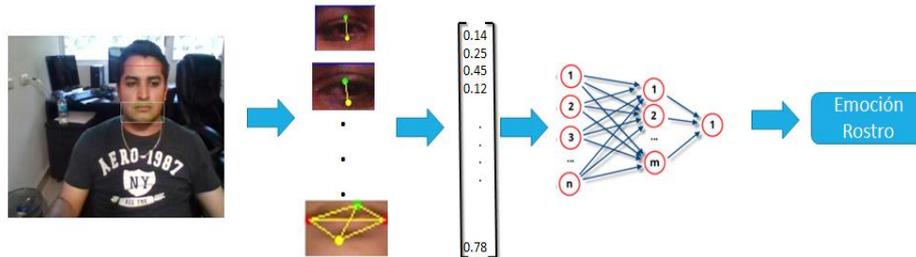


Fig. 2. Algoritmo del reconocedor de emociones en rostros.

- a) Recibe una imagen.
b) Detecta el rostro en la imagen con OpenCV.
c) Repite los siguientes pasos para boca, ojo izquierdo, ojo derecho, ceja izquierda, ceja derecha, lo llamaremos objetivo:
– Ubicar dentro del rostro posible zona de ubicación del objetivo, se llamará región de interés.
– Dentro de la región de interés aplicar un método de búsqueda en cascada de OpenCV para buscar en concreto el objetivo que estamos buscando.
– Ya con el objetivo encontrado, se le aplican filtros de OpenCV [29] como se muestra en la Figura 3 para determinar los puntos clave en el rostro.
d) Se crea un vector de características de 10 elementos, donde cada uno es una distancia euclidiana entre diferentes puntos clave ubicados en el rostro.
e) Se ingresa el vector de características a la red neuronal ya entrenada para que lo clasifique a una de las emociones.

f) Regresa la emoción obtenida.

Este reconocedor regresa los valores de emociones para feliz, triste, enojado, sorpresa y neutral.

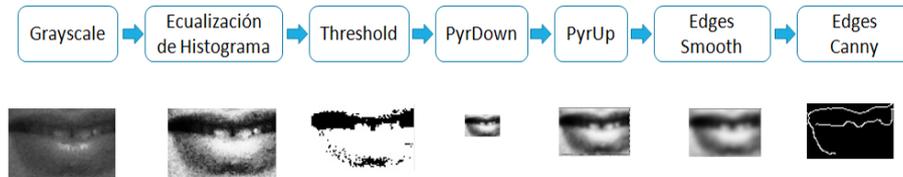


Fig. 3. Filtros de OpenCV en la boca.

1.1.3. Reconocedor de emociones en voz

El reconocedor de emociones en voz que utiliza PREMOC identifica la valencia del audio recibido (positiva o negativa) y también consiste en dos fases: entrenamiento y ejecución.

En la fase de entrenamiento se utilizaron un total de 45 audios de 9 sujetos diferentes, donde cada audio está clasificado en positivo o negativo. De los audios se extraen las características utilizando su espectrograma con la librería musicg [30] para obtener los parámetros para entrenar una SVM de la librería LibSVM [31]. La instancia de la SVM ya entrenada es guardada para utilizarla online en la clasificación de sonidos de PREMOC.

La fase de ejecución en línea funciona de la siguiente manera y como se muestra en la Figura 4.



Fig. 4. Algoritmo del reconocedor en voz.

- a) Se recibe un audio en formato *wav* serializado en Base64.
- b) Se obtienen el espectrograma del audio para obtener los valores de tono e intensidad a lo largo del audio.
- c) Se calculan los siguientes valores:
 - El valor de la intensidad más alta del audio.
 - El valor de la intensidad más baja del audio.
 - El valor del tono más alto del audio.
 - El valor del tono más bajo del audio.
- d) Estos 4 valores se introducen a la SVM para obtener la valencia de la voz.
- e) Se regresa la valencia obtenida.

Este reconocedor regresa la valencia de la voz (*positiva o negativa*).

1.1.4. Reconocedor de emociones en señales EEG

El reconocedor de emociones por medio de señales EEG también consta de dos fases: entrenamiento e implementación en línea. Este reconocedor identifica la valencia de la emoción (positivo, negativo o neutral) del archivo recibido de señales EEG. El reconocedor fue creado y probado para la Interface Cerebro-Computadora (BCI) Emotiv Epoc [32]. Para seleccionar los canales apropiados que necesitamos del BCI, se consideró el trabajo de Mahajan [33] que usa los canales frontales AF3, AF4, F3, F4, F5 Y FC6. También el trabajo de Liu [6] que usa los canales AF3, F4 Y FC6. Al final se consideraron los canales AF3, F3, F4, FC5 y FC6 porque se concluyó que son los que más se aproximan a las emociones humanas.

La característica principal extraída de las señales EEG es el Exponente de Hurst, el cual es usado en análisis de series de tiempo para identificar un comportamiento inestable de las señales EEG para identificar tendencias en los datos. Se eligió el Exponente de Hurst por los buenos resultados mostrados en Wang [34](71.38 %).

La fase de entrenamiento esta descrita en [35]. Y la fase en línea funciona como muestra la Figura 5 y como se describe a continuación.

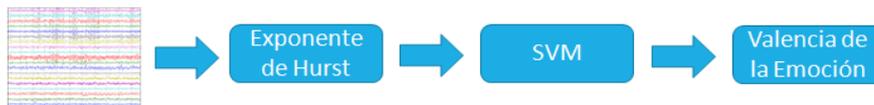


Fig. 5. Reconocedor de señales EEG.

- PREMOC recibirá un archivo edf con las señales EEG.
- Se extraerán la información solamente de los 5 canales mencionados anteriormente.
- Se calculará el Exponente de Hurst de cada canal para obtener un vector de tamaño 5.
- Alimentamos la SVM de la librería LibSVM [31] con el vector obtenido y se obtendrá la valencia de la emoción.

Este reconocedor regresa la valencia de la emoción: positiva, negativa o neutral.

1.2. Integración de las salidas de los reconocedores

Para mayor control sobre lo que el usuario requiere, la plataforma ofrece 3 tipos de modo de respuesta, que son: *simple*, *multi* y *student*. Cada uno de ellos hace uso de los reconocedores individuales descritos en la sección anterior para determinar sus respectivas salidas.

1.2.1. Modo simple

En el modo simple, la respuesta que ofrece PREMOC es la que da cada reconocedor por separado sin integrarlos; es decir, si un usuario requiere la emoción en un rostro y en un sonido de voz, PREMOC le da en respuesta la emoción del rostro y la emoción de la voz. Las emociones que regresa el modo simple dependen de cada reconocedor individual.

Como se muestra en la Figura 6, el integrador reúne las respuestas de los reconocedores individuales para crear el json de respuesta para el usuario.

1.2.2. Modo multi

En el modo multi, la respuesta que da el usuario es multimodal; es decir, regresa una sola emoción al usuario de los dispositivos que él requiera. En el mismo caso que el ejemplo anterior, si el usuario requiere la emoción en un rostro y en un sonido de voz y lo quiere en forma multimodal, PREMOC le da en respuesta una emoción obtenida después de analizar mediante reglas difusas la emoción que obtuvo desde el reconocedor de rostro y desde el reconocedor de voz. Las emociones que se generan con las reglas difusas son las mismas: feliz, sorpresa, neutral, enojado y triste.

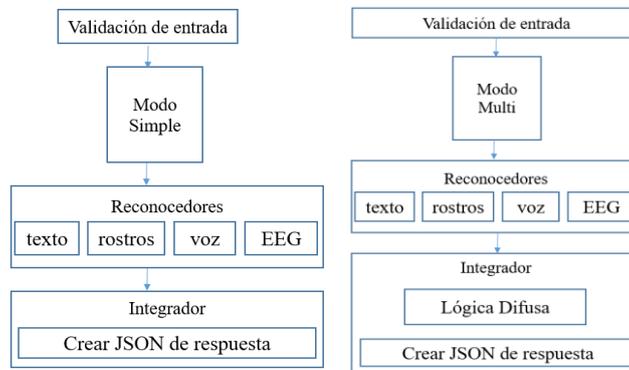


Fig. 6. Funcionamiento de Modo Simple y Modo Multi.

Como muestra la Fig. 6, en modo multi, una vez aplicados los reconocedores individuales, a diferencia del modo simple, utiliza un motor difuso que consta de una serie de reglas difusas para determinar una sola emoción considerando en cuenta dos o más.

Se creó un generador de reglas difusas para la creación de reglas en todas las combinaciones de dispositivos que pudieran darse. El generador de reglas está basado en la tabla de Whissell (Figura 7). Se ubica a cada emoción de los diferentes dispositivos en la tabla y se calcula un promedio para determinar la emoción final.

La tabla de Whissell está basada en activación y evaluación, y están distribuidas en ella las emociones.

Table 1. Ubicación de las emociones en la tabla de Whissell.

Emoción	Punto
Feliz	(2.2 , 1)
Sorpresa	(1.5 , 2)
Neutral	(0 , 0)
Enojado	(-0.5 , 1.1)
Triste	(-1.5 , -0.5)

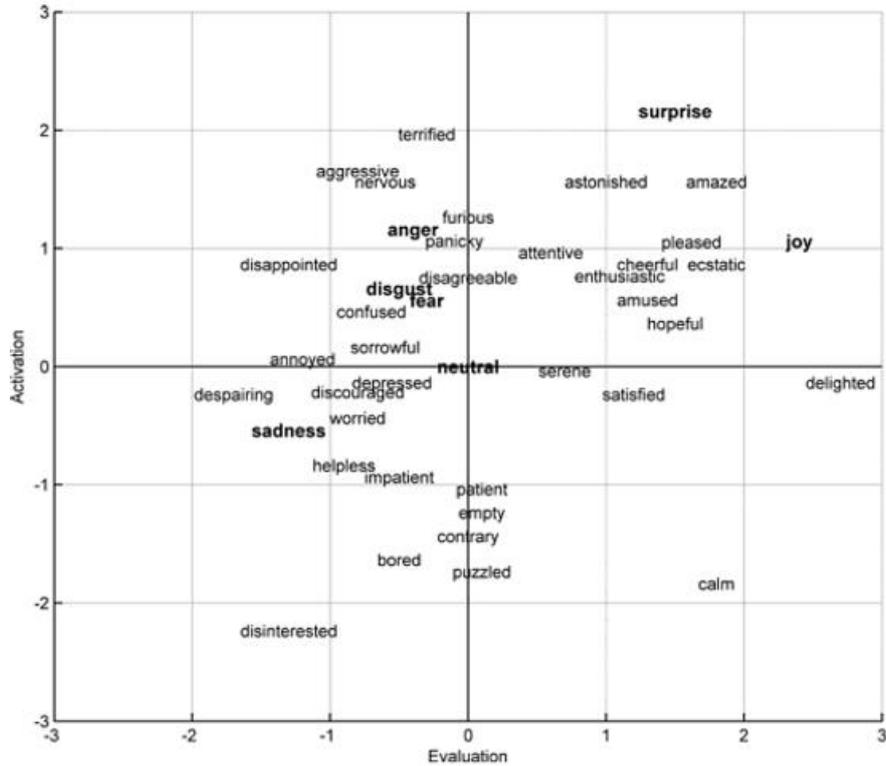


Fig. 7. Tabla de Whissell.

Los reconocedores de voz y de señales EEG solo regresan la valencia (emoción positiva y negativa), y también emoción neutral en el caso de señales EEG. Tomando en cuenta esto y para la integración de estas señales con el resto (expresiones faciales y texto), se realizaron diferentes pruebas para determinar una ubicación promedio de una valencia positiva o negativa en la tabla de Whissell y se determinó ubicarlas en los puntos que muestra la tabla 2.

Table 2. Ubicación de positivo y negativo de wav y EEG en la tabla de Whissell.

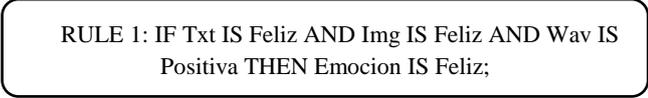
Emoción	Punto
Positivo	(1,0.8)
Negativo	(-0.6,0.1)

Con estos valores definidos, ya se pueden ubicar todos los resultados de los cuatro reconocedores dentro de la tabla de Whissell. El procedimiento para obtener una emoción multimodal en PREMOC es:

- Ubicar las emociones de los reconocedores individuales en la tabla de Whissell.
- Obtener un punto promedio entre las ubicaciones de las emociones.
- Calcular la distancia desde el punto promedio hasta la ubicación de las emociones de la tabla 1.

d) Seleccionar la emoción que tenga la distancia menor al punto promedio.

De esta manera se selecciona la emoción final para el modo multimodal en el generador de reglas; las reglas son creadas con el formato de la regla que se muestra en la Figura 8, y son procesadas por la librería jFuzzyLogic [36]. De esta manera se crean todas las reglas difusas necesarias para la integración de emociones en el modo multimodal. En total fueron creadas 470 reglas para las combinaciones de estos 4 reconocedores.



```
RULE 1: IF Txt IS Feliz AND Img IS Feliz AND Wav IS
Positiva THEN Emocion IS Feliz;
```

Fig. 8. Regla difusa.

1.2.3. Modo student

En modo student, el usuario obtendrá como respuesta el valor de una emoción tomando en cuenta las que normalmente son comunes en un ámbito de aprendizaje. Estas emociones son enganchado, aburrido, frustrado y neutral.

La creación de reglas difusas del modo student fue similar al modo multi, usando la tabla de Whissell. El modo student regresa al usuario las emociones educativas enganchado, frustrado, aburrido y neutral.

4. Experimentos

Actualmente PREMOC está siendo utilizado en varios proyectos dentro del laboratorio y entre ellos tenemos a EasyLogic y STAAM; ambos son Sistemas Tutores Inteligentes y Afectivos, donde el reconocimiento de emociones es realizado con PREMOC.

EasyLogic es un sistema web, que actúa como Sistema Tutor Inteligente Afectivo en el proceso de aprendizaje de programación de un lenguaje. EasyLogic utiliza el reconocimiento de emociones en rostro de PREMOC en modo student cada 15 segundos, para mostrar al estudiante dos tipos de ayudas mientras aprende lógica algorítmica: informativas y motivacionales (figura 10).

STAAM es un Sistema Tutor Inteligente Afectivo para matemáticas para niños de primer año de secundaria. STAAM utiliza el reconocedor de emociones en rostros y el reconocedor de emociones en texto de PREMOC para calificar los ejercicios resueltos por los alumnos. STAAM considera el estado afectivo del estudiante además del tiempo y la resolución del ejercicio y entrega un trofeo diferente al estudiante dependiendo del estado de ánimo con el que resolvió el ejercicio.

5. Conclusiones y trabajo futuro

Como conclusión, el reconocedor de emociones en imágenes según pruebas realizadas tiene una tasa de éxito del 80%. El reconocedor de emociones en texto tiene poco más de 80% según pruebas realizadas con estudiantes. El reconocedor de

emociones en voz obtuvo 64%. El reconocedor de emociones en señales EEG muestra su tasa de éxito en 70%. Las pruebas fueron realizadas con los mismos datos de entrenamiento.

Se puede concluir que tener un sistema de reconocimiento multimodal de emociones en línea, facilita la integración de afecto a los desarrolladores, ya que pueden crear su software sin realizar el entrenamiento de reconocedores de emociones.

Actualmente se está trabajando en mejorar el reconocedor de emociones en rostros con diferentes técnicas de extracción de características como: Action Units [37] y Binary Pattern [38]. Como trabajo futuro se considera la mejora del reconocedor de emociones en voz con más datos de entrenamiento y que no sea de solo valencia, sino, de las emociones *feliz*, *enojado*, *triste* y *neutral*. También se considera mejorar el reconocedor de texto y de señales EEG con otros tipos de extractores de características. También integrar el reconocimiento de otros dispositivos el rastreo de ojos.

Referencias

1. Picard, R.W.: Affective Computing. No. 321, pp. 1–16 (1995)
2. Mehrabian, A.: Communication without words. *Psychol. Today*, Vol. 2, pp. 53–56 (1968)
3. Ekman, P.: Strong Evidence for Universals in Facial Expressions: A Reply to Russell's Mistaken Critique. *Psychol. Bull.*, Vol. 2, No. 1, pp. 268–287 (1994)
4. Han, K., Yu, D., Tashev, I.: Speech Emotion Recognition Using Deep Neural Network and Extreme Learning Machine. pp. 223–227 (2014)
5. Seol, Y.S., Kim, D.J., Kim, H.W.: Emotion Recognition from Text Using Knowledge-based ANN. *Proceedings of 23rd International Technical Conference on Circuits/Systems, Computers and Communications*, pp. 1569–1572 (2008)
6. Liu, Y., Sourina, O., Nguyen, M.K.: Real-time EEG-based Human Emotion Recognition and Visualization. *Trans. Comput. Sci. XII*, Vol. 6670, Special Issue on Cyberworlds, pp. 256–277 (2011)
7. Shaheen, S., El-Hajj, W., Hajj, H., Elbassuoni, S.: Emotion Recognition from Text Based on Automatically Generated Rules. *IEEE Int. Conf. Data Min. Work.*, pp. 383–392 (2014)
8. Zheng, D., Tian, F., Liu, J., Zheng, Q., Qin, J.: Emotion Chat: A Web Chatroom with Emotion Regulation for E-Learners. *Phys. Procedia*, Vol. 25, pp. 763–770 (2012)
9. Khandait, S.P., Thool, R.C., Khandait, P.D.: Automatic Facial Feature Extraction and Expression Recognition based on Neural Network. *Int. J. Adv. Comput. Sci. Appl.*, Vol. 2, No. 1, pp. 113–118 (2011)
10. Gangwar, S., Shukla, S., Arora, D.: Human Emotion Recognition by Using Pattern Recognition Network. *Int. J. Eng. Res. Appl.*, Vol. 3, No. 5, pp. 535–539 (2013)
11. De, A., Saha, A., Pal, M.C.: A Human Facial Expression Recognition Model Based on Eigen Face Approach. *Procedia Comput. Sci.*, Vol. 45, pp. 282–289 (2015)
12. Kostoulas, T., Fakotakis, N.: A speaker dependent emotion recognition framework. *Proc. 5th Int. Symp. Commun. Syst. Networks Digit. Signal Process.*, University of Patras, pp. 305–309 (2006)
13. Vogt, T., André, E., Bee, N.: *EmoVoice — A Framework for Online Recognition of Emotions from Voice. Perception in Multimodal Dialogue Systems*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 188–199 (2008)
14. Caballero-Morales, S.O.: Recognition of emotions in Mexican Spanish speech: An approach based on acoustic modelling of emotion-specific vowels. *Sci. World J.*, Vol. 2013, p. 13 (2013)
15. Nie, D., Wang, X.W., Shi, L.C., Lu, B.L.: EEG-based emotion recognition during watching movies. *5th Int. IEEE/EMBS Conf. Neural Eng. NER*, pp. 667–670 (2011)

16. Lee, Y.Y., Hsieh, S.: Classifying different emotional states by means of eegbased functional connectivity patterns. *PLoS One*, Vol. 9, No. 4, pp. 1–13 (2014)
17. Alonso-Martín, F., Malfaz, M., Sequeira, J., Gorostiza, J.F., Salichs, M.A. A multimodal emotion detection system during human-robot interaction. *Sensors (Basel)*, Vol. 13, No. 11, pp. 15549–15581 (2013)
18. Busso, C., Deng, Z., Yildirim, S., Bulut, M., Lee, C.M., Kazemzadeh, A., Lee, S. Neumann, U., Narayanan, S.: Analysis of emotion recognition using facial expressions, speech and multimodal information. 6th Int. Conf. Multimodal interfaces, pp. 205–211, (2004)
19. Chuang, Z.J., Wu, C.: Multi-modal emotion recognition from speech and text. *J. Comput. Linguist. Chinese*, Vol. 9, No. 2, pp. 45–62 (2004)
20. D’Mello, S.K., Graesser, A.: Multimodal semi-automated affect detection from conversational cues, gross body language, and facial features. *User Model. User-Adapted Interact.*, Vol. 20, No. 2, pp. 147–187 (2010)
21. Caridakis, G., Castellano, G., Kessous, L., Raouzaïou, A., Malatesta, L., Asteriadis, S., Karpouzis, K.: Multimodal emotion recognition from expressive faces, body gestures and speech. *IFIP Int. Fed. Inf. Process.*, Vol. 247, pp. 375–388 (2007)
22. Paredes, P., Berkeley, U.C., Sun, D.: Sensor-less Sensing for Affective Computing and Stress Management Technology. *Pervasive Comput. Technol. Healthc. Work.*, pp. 459–463 (2013)
23. Wagner, J., Andre, E., Jung, F.: Smart sensor integration: A framework for multimodal emotion recognition in real-time. 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops, pp. 1–8 (2009)
24. Gonzalez-Sanchez, J., Chavez-Echeagaray, M.E., Atkinson, R., Bursleson, W.: ABE: An agent-based software architecture for a multimodal emotion recognition framework. Proc. - 9th Work. IEEE/IFIP Conf. Softw. Archit. WICSA, April 2016, pp. 187–193, (2011)
25. Zatarain Cabada, R., Barron Estrada, M.L., Hernandez, F.G., Bustillos, R.O.: An Affective Learning Environment for Java. IEEE 15th International Conference on Advanced Learning Technologies, pp. 350–354 (2015)
26. Zatarain-Cabada, R., Barrón-Estrada, M.L., García-Lizárraga, J., Muñoz-sandoval, G., Rios-Feliz, J.M.: Java Tutoring System with Facial and Text Emotion Recognition. *Res. Comput. Sci.*, Vol. 106, No. April 2016, pp. 49–58 (2015)
27. Langner, O., Dotsch, R., Bijlstra, G., Wigboldus, D.H.J., Hawk, S.T., van Knippenberg, A.: Presentation and validation of the Radboud Faces Database. *Cogn. Emot.*, Vol. 24, No. 8, pp. 1377–1388, (2010)
28. Witten, I. H., Frank, E., Trigg, L., Hall, M., Holmes, G., Cunningham, S.J.: *Weka : Practical Machine Learning Tools and Techniques with Java Implementations*. Seminar, Vol. 99, pp. 192–196 (1999)
29. OpenCV Filtros. [Online]. Available: <http://docs.opencv.org/2.4/modules/imgproc/doc/imgproc.html>.
30. Musicg, Spectrogram. [Online]. Available: <http://deeplearning4j.org/canovadoc/musicg/extension/Spectrogram.html>.
31. Chang, C., Lin, C.: LIBSVM : A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.*, Vol. 2, pp. 1–39 (2011)
32. Inc, E., Epoc - Emotiv. [Online]. Available: <https://emotiv.com/epoc.php>.
33. Mahajan, R., Bansal, D., Singh, S.: A Real Time Set Up for Retrieval of Emotional States from Human Neural Responses. *World Acad. Sci. Eng. Technol.*, Vol. 8, No. 3, pp. 144–149 (2014)
34. Wang, X.-W., Nie, D., Lu, B.-L.: Emotional state classification from EEG data using machine learning approach. *Neurocomputing*, Vol. 129, pp. 94–106 (2014)
35. Barrón-Estrada, M.L., Zatarain-Cabada, R., Aispuro-Gallegos, C.L., Sosa-Ochoa, C.G. Lindor-Valdez, M.: Affective Environment for Java Programming Using Facial and EEG Recognition. *Res. Comput. Sci.*, Vol. 106, pp. 39–47 (2015)

36. Cingolani, P., Alcalá-Fdez, J.: jFuzzyLogic: a Java Library to Design Fuzzy Logic Controllers According to the Standard for Fuzzy Control Programming. *Int. J. Comput. Intell. Syst.*, Vol. 6, No. sup1, pp. 61–75 (2013)
37. Tian, Y.-L., Kanade, T., Cohn, J.F.: Recognizing upper face action units for facial expression analysis. *Proc. IEEE Conf. Comput. Vis. Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, Vol. 1, No. 2, pp. 1–19 (2000)
38. Feng, X., Pietikäinen, M., Hadid, A.: Facial Expression Recognition with Local Binary Patterns and Linear Programming. *Pattern Recognit. Image Anal.*, Vol. 15, No. 2, pp. 546–548 (2005)

Sistema tutor afectivo para la enseñanza de lógica algorítmica y programación

Ramón Zatarain-Cabada¹, María Lucía Barrón-Estrada¹,
José Mario Ríos-Félix¹, Giner Alor-Hernandez²

¹Instituto Tecnológico de Culiacán, Culiacán Sinaloa,
México

²Instituto Tecnológico de Orizaba,
División de Estudios de Posgrado e Investigación, Orizaba, Veracruz,
México

{rzatarain, lbarron, mario_rios}@itculiacan.edu.mx, galor@itorizaba.edu.mx

Resumen. La creciente demanda de herramientas de software que motiven y apoyen a los estudiantes en el aprendizaje de diseño e implementación de algoritmos y programas, ha motivado la creación de este tipo de sistemas de software. En este artículo presentamos un nuevo e innovador sistema tutor afectivo de lógica algorítmica y programación, basado en la técnica de bloques. Nuestro enfoque combina la interfaz de Google Blockly con técnicas de gamificación y ejercicios que son monitoreados para obtener el estado afectivo del estudiante. Dependiendo de la emoción manifestada (aburrido, enganchado, frustrado y neutral), el sistema evalúa una serie de variables, para determinar si el estudiante requiere asistencia. En base a las pruebas preliminares con varios estudiantes, hemos obtenido resultados satisfactorios, que indican que nuestro sistema contribuye de forma adecuada al proceso de enseñanza de lógica algorítmica.

Palabras clave: Computación afectiva, sistemas tutores inteligentes, lógica algorítmica, lenguajes de programación.

Affective Tutoring System for Teaching Algorithmic Logic and Programming

Abstract. The growing demand for software tools that encourage and support students in learning design and algorithm implementation, has allowed the creation of such software systems. In this paper we present a new and innovative affective tutoring system, for logic and algorithmic programming, based on block techniques. Our approach combines the Google Blockly's interface with gamification techniques and exercises that are monitored according to the emotional state of the student. Depending on the expressed emotion (boring, engagement, frustration and neutral), the system evaluates a number of variables to determine whether the student requires assistance. Based on preliminary tests

with several students, we have obtained satisfactory results, which indicate that our system properly contributes to the process of learning algorithm logic.

Keywords: Affective computing, intelligent tutoring systems, algorithmic logic, programming languages.

1. Introducción

Aprender a programar sigue siendo una de las tareas más demandantes para la mayoría de los estudiantes. Cada año se imparten cursos universitarios sobre programación básica a nuevas generaciones de estudiantes. Incluso carreras ajenas a las ciencias de la computación incluyen en su plan de estudios materias de programación básica.

En la actualidad existen muchas herramientas, que pretenden apoyar a los estudiantes, en su ardua tarea de aprender temas relacionados con la computación, debido a que la ciencia de la computación no es sólo acerca de la programación, sino toda una forma de pensar [1]. Podemos clasificar estas herramientas según su enfoque. Algunas de estas herramientas de software son los sistemas llamados e-learning o CAI (Computer Assisted Instruction), los sistemas tutores inteligentes (STI) de programación y los juegos serios que instruyen los temas de programación.

Las herramientas o sistemas e-learning de apoyo para la programación pretenden facilitar el aprendizaje, haciendo uso de marcos de trabajo pre-establecidos, que les facilitan la implementación de los algoritmos y programas a los estudiantes. Dentro de esta clasificación, existen las herramientas de autor, las cuales permiten a los usuarios diseñar y construir sus propias historias interactivas, animaciones, simulaciones o incluso juegos. Dentro de las herramientas de autor, destacan aquellas que pretenden instruir a los estudiantes en lo que se conoce como “Pensamiento Computacional”. El Pensamiento Computacional es una serie de habilidades que integran la resolución de problemas, el diseño de sistemas y la comprensión de la conducta humana que se basa en los conceptos fundamentales de la computación [2].

Por otra parte, los STI son ambientes computarizados de aprendizaje que incorporan los modelos computacionales de las ciencias cognitivas, ciencias de aprendizaje, lingüística computacional, inteligencia artificial entre otros campos. Un STI brinda seguimiento de los estados psicológicos de los alumnos en los detalles finos, un proceso llamado modelado de los estudiantes [3]. Dentro de la familia de los STI, existen los Sistemas Tutores Afectivos (STA), quienes son sistemas inteligentes que incorporan la capacidad de reconocer el estado emocional de los estudiantes, con lo que permiten al usuario interactuar con ejercicios que estimulen su estado emocional.

Finalmente, los juegos serios son aquellos juegos que más que entretener, pretenden enseñar a través de un juego. De esta forma atraen y retienen a los estudiantes más fácilmente que otras herramientas de enseñanza [4]. De la idea de los juegos, surge lo que se conoce como gamificación. Básicamente, la gamificación es un término que se utiliza para describir aquellas características de un sistema interactivo que tienen como objetivo motivar y comprometer a los usuarios finales a través del uso y la mecánica de estímulos comúnmente incluidos en los juegos, como por ejemplo, los trofeos, niveles y las tablas de posiciones [5].

EasyLogic es una herramienta que combina las tecnologías previamente descritas (herramientas e-learning para enseñanza de programación, sistemas tutores inteligentes y afectivos y gamificación) que permite a un estudiante diseñar e implementar algoritmos de una forma simple pero a su vez poderosa. La principal contribución de la herramienta y de este trabajo, es que incluye aspectos afectivos y motivacionales por primera vez dentro de una herramienta que enseña a un estudiante el difícil proceso de construcción de algoritmos y programas.

Este artículo está organizado de la siguiente forma: la Sección 2 describe los trabajos relacionados. La Sección 3 presenta la estructura y funcionamiento de la herramienta (EasyLogic). En la Sección 4 se muestra una sesión de trabajo con EasyLogic. La Sección 5 contiene algunos de los experimentos realizados y los resultados obtenidos. Por último en la Sección 6 se describen las conclusiones y trabajos futuros.

2. Trabajos relacionados

En esta sección se describen investigaciones y trabajos en áreas relacionadas con nuestra investigación.

2.1. Herramientas e-learning de programación

Dentro de las herramientas de programación más importantes podemos mencionar a GreenFoot [6], un ambiente de desarrollo integrado, que combina la programación en Java con una interfaz gráfica interactiva. Greenfoot se enfoca a facilitar la enseñanza de temas de programación orientada a objetos. Otra herramienta importante es Alice [7], en donde los estudiantes novatos pueden programar juegos en 3D utilizando modelos de objetos ya predefinidos.

Por otra parte, en los últimos años se han desarrollado herramientas que permiten aprender programación para móviles, como es el caso de MIT App Inventor [8], que permite diseñar y construir aplicaciones móviles para Android totalmente funcionales. Con esta herramienta los desarrolladores se enfocan en la lógica y el algoritmo de solución, en lugar de la sintaxis del lenguaje de programación.

2.2. Herramientas de autor

Algunas de las herramientas de autor que abordan conceptos de pensamiento algorítmico y programación básica son: Scratch [9], [10], una comunidad de aprendizaje creativo en línea, donde los usuarios pueden interactuar y subir sus proyectos para favorecer el reúso y permitir la remezcla de ellos. Para crear un proyecto se emplean bloques gráficos, donde cada bloque representa un elemento del lenguaje de programación, como son: estructura de control, operadores, variables, funciones, etc.

Otro ejemplo es Scalable Game Design Arcade (AgentSheets) [1], [11], que permite crear juegos de manera fácil, para lo cual utiliza agentes pedagógicos, empleando técnicas de gamificación. También, es importante mencionar a Code Studio [12], el cual se presenta mediante una página principal con los cursos en línea creados por

Code.org, y que representa la comunidad donde un estudiante puede aprender diferentes temas de ciencias de la computación.

Por último, PseInt [13] es una herramienta para asistir a un estudiante mediante un simple e intuitivo pseudo-lenguaje en español, complementado con un editor de diagramas de flujo que permite centrar la atención en los conceptos fundamentales de lógica de programación y al uso de estructuras de control.

2.3. Sistemas tutores inteligentes y afectivos

Nuestro principal interés son aquellos tutores inteligentes que realizan una intervención dependiendo del estado afectivo del estudiante.

Entre los STI afectivos más novedosos se encuentran Gaze tutor, un tutor inteligente para el área de biología que mediante un rastreo de los ojos de los estudiantes detecta si ellos se encuentran aburridos, enganchados/motivados o distraídos. El tutor utiliza diálogos textuales a través de un agente pedagógico para tratar de reorientar la atención del estudiante [14]. Por otra parte Gerda es un ITS para temas de sistemas operativos, el cual mediante diálogos (preguntas y respuestas), analiza el estado afectivo del estudiante utilizando diccionarios de palabras con anotaciones afectivas [15].

Tal vez el más importante de los STI afectivos es Affective AutoTutor, una extensión de AutoTutor (un ITS que ayuda a los estudiantes a aprender contenido técnico complejo en diferentes áreas del conocimiento), que agrega capacidades de reconocimiento y manejo de emociones mediante la detección del estado afectivo del estudiante, utilizando expresiones faciales, sensores de movimiento del cuerpo y conversación textual [16].

2.4. Juegos serios

Dos de los juegos más interesantes son Program your robot [17] y Cargo-bot [18]. El primero se enfoca a la enseñanza de pensamiento algorítmico y el segundo plantea una nueva metodología de enseñanza de la técnica de recursión a través de un juego. Ambos son juegos muy bien logrados, que atrapan a los usuarios, quienes no se dan cuenta que están aprendiendo mientras juegan.

2.5. Panorama actual

Una de las principales debilidades de esta clase de herramientas enfocadas al aprendizaje de temas relacionados con el pensamiento computacional, y en específico lógica algorítmica, es que solo se orientan a estudiantes de nivel pre-universitario (primaria y secundaria principalmente), y no contemplan el estado afectivo de los estudiantes cuando estos las usan. Immordino-Yang and Damasio establecieron que los procesos emocionales actúan como un "timón" en la transferencia de conocimientos y habilidades, y que las emociones juegan un papel importante y necesario en la toma de decisiones, ya que es la interfaz entre la cognición que conduce a desarrollar potencialmente la creatividad [19].

Además, como establecen J. Maloney y M. Kölling, acerca de Scratch y Greenfoot, la filosofía que utilizan para enseñar es que al estudiante primero debes dejarlo jugar,

permitirle crear algo, motivarlo a ser creativo y luego explicarle lo que realmente está pasando. Ellos concluyen que una persona que utilice Scratch puede continuar su enriquecimiento de habilidades por medio de Greenfoot y luego migrar hacia algún IDE (Scratch → Greenfoot → IDE) [20].

Hasta la fecha, no existe un sistema integral, con el que el estudiante pueda aprender y dominar los elementos de lógica algorítmica y programación de forma progresiva, y que además tome en cuenta el estado afectivo del mismo estudiante. Para poder manejar este gran reto, la herramienta deberá manejar 3 etapas:

1. Fungir como tutor para desarrollar un pensamiento computacional.
2. Permitir practicar los conocimientos de lógica algorítmica.
3. Guiar al estudiante hacia una transición a un lenguaje de programación.

La contribución principal de EasyLogic, es contar con un STI para la enseñanza de lógica algorítmica a nivel universitario que además de utilizar técnicas de gamificación, permita reconocer y manejar el estado afectivo de los estudiantes, como apoyo a elementos cognitivos usados en la individualización del proceso de enseñanza/aprendizaje de los estudiantes.

3. EasyLogic

EasyLogic es un sistema web, que actúa como STI afectivo en el proceso de aprendizaje de programación de un lenguaje. Nuestro sistema se conforma de 3 secciones principales: a) Aprende. Se proveen diversos cursos, que incluyen una serie de ejercicios, para dar lección a cada una de las distintas estructuras de control que existen. Estos ejercicios son supervisados para obtener el estado afectivo que el estudiante está presentando. b) Imagina y crea. Esta sección permite a los estudiantes diseñar sus propios algoritmos y posteriormente ejecutarlos. Asimismo puede generar código en un lenguaje de programación asociado al algoritmo elaborado. c) Codifica. En esta sección se puede programar directamente en JavaScript en lugar de usar los bloques gráficos (aún en desarrollo).

Al igual que muchas herramientas de autor y de programación modernas, EasyLogic utiliza bloques gráficos. Nuestra interfaz se logra utilizando la librería de Google Blockly [21]. Blockly es una librería de JavaScript que además de ser libre, es personalizable y extensible, por lo que se crearon nuevos bloques adecuados para los diversos ejercicios.

3.1. PREMOC

Para el reconocimiento emocional se utilizó una Plataforma de Reconocimiento Multimodal de Emociones (PREMOC), la cual se está desarrollando en el laboratorio de Maestría en Ciencias de la Computación del Instituto Tecnológico de Cuiacán. PREMOC Consiste en un servicio web que puede utilizar: imágenes faciales, audio, texto y señales electroencefalografías (EEG), las cuales procesa para brindar como resultado una o varias emociones, dependiendo del modo de integración a utilizar.

PREMOC cuenta con 3 modos de integración: Simple, Multi y Student. En el modo Simple, la respuesta enviada, es la emoción detectada por cada reconocedor

individualmente, sin integrarlos. Por ejemplo, en caso de enviar datos de una imagen facial, y datos de señales EEG, la respuesta serán 2 emociones independientes, una para la imagen facial y otra para la señal EEG. En el modo Multi, la respuesta al usuario es multimodal, es decir, envía una sola emoción al usuario, integrando todas las emociones detectadas por cada reconocedor. Esto se logra utilizando un integrador que se implementó en lógica difusa. En el modo Student, a diferencia de los modos de integración anteriores, que utilizan las emociones de Ekman, el usuario obtendrá una respuesta multimodal orientada a la educación (enganchado, aburrido, frustrado y neutral). La Figura 1 muestra la estructura y funcionamiento de PREMOC.

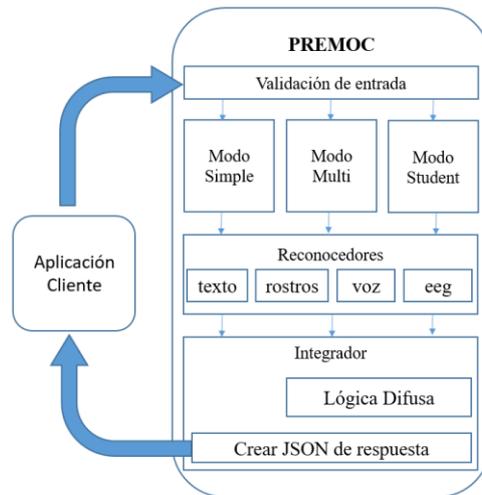
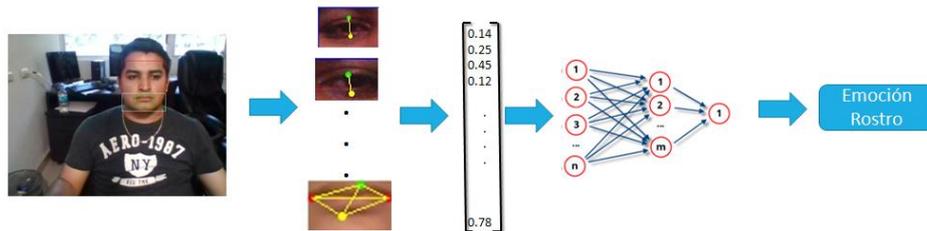


Fig. 1. Estructura de PREMOC.

Para la versión actual del sistema, se utiliza el modo Student, y se envía información de imágenes faciales. El reconocedor de imágenes de PREMOC consiste en dos fases: el Entrenamiento y la Ejecución. La fase de entrenamiento consistió en entrenar la red neuronal encargada de la clasificación de imágenes. Para esto se tomaron como datos de entrenamiento las imágenes del corpus RaFD [22], del cual se utilizaron 955 imágenes. De cada una de las imágenes se extrajo un vector de 10 características, donde cada característica es una distancia euclidiana entre diferentes puntos del rostro. Con estos datos se entrenó una red neuronal de propagación hacia atrás, empleando la



librería Weka para Java. La fase de ejecución se describe en la Figura 2.

Fig. 2. Algoritmo reconocedor de emociones, por medio de imagen facial.

3.2. Ayudas

Una vez que el usuario procede a realizar un ejercicio, el sistema comienza a obtener imágenes del usuario, para ser enviadas de forma continua a PREMOC. Luego de ser analizadas las imágenes, se obtiene una emoción (aburrido, enganchado, confundido, neutral); posteriormente se almacena en el sistema la emoción presentada por el usuario. Cada 30 segundos, EasyLogic evalúa diversas variables como el tiempo invertido en el ejercicio, cantidad de ejecuciones y la emoción que más se ha detectado recientemente por el usuario, para determinar si el estudiante necesita de algún tipo de ayuda.

Las ayudas consisten en ventanas emergentes que se muestran automáticamente cuando se detecta que el usuario las necesita. Las ayudas se dividen en tres tipos: iniciales, informativas y motivacionales. En la Figura 3 se describen cada una de las ayudas utilizadas por EasyLogic.

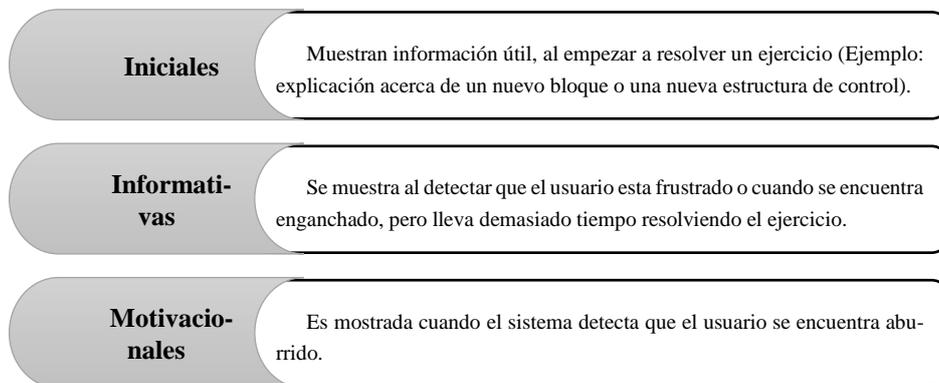


Fig. 3. Tipos de ayudas utilizadas en EasyLogic.

EasyLogic implementa técnicas de gamificación para motivar a los estudiantes, otorgándoles puntos y trofeos conforme van avanzando en su curso. Debido a que la herramienta pretende usarse en cursos reales, los estudiantes deberán seleccionar el grupo al que pertenecen al momento de registrarse. Cada estudiante podrá visualizar el puntaje que cada uno de sus compañeros tiene hasta el momento y los trofeos que han logrado obtener. Se pretende con esto estimular a que sean competitivos y se enganchen en la herramienta.

4. Una sesión de trabajo con EasyLogic

El sistema inicia en una pantalla de bienvenida, donde se muestran información sobre los cursos, acceso al área libre para la creación de algoritmos y acceso a la sección de programación. Para acceder a los cursos, los usuarios deberán iniciar sesión utilizando su correo electrónico y contraseña. Una vez seleccionado el curso el sistema cargará automáticamente el siguiente ejercicio aún sin resolver por el estudiante. Algunos ejercicios están asociados a ayudas iniciales; una vez cargado el ejercicio se

muestra la ayuda inicial en caso de requerirse. Para no aburrir al estudiante con ayudas que ya ha leído anteriormente, se guarda información de todas aquellas ayudas que se le han mostrado al estudiante y se ha confirmado que la información ha sido leída.

Una vez cargado el ejercicio, el sistema deja trabajar al estudiante por un tiempo específico. En caso de resolver el ejercicio se calculan los puntos obtenidos en base al tiempo invertido en el ejercicio y a la cantidad de ejecuciones requeridas. En caso de obtener un nuevo trofeo se muestra al estudiante. En caso de no haber resuelto el ejercicio, se analizan los datos del usuario (tiempo transcurrido, estado emocional, cantidad de ejecuciones) para verificar si el usuario necesita algún tipo de ayuda extra (informativa o motivacional). De ser así se obtiene de una base de datos una ayuda para el ejercicio actual. En la Figura 4 se muestra el proceso descrito anteriormente.

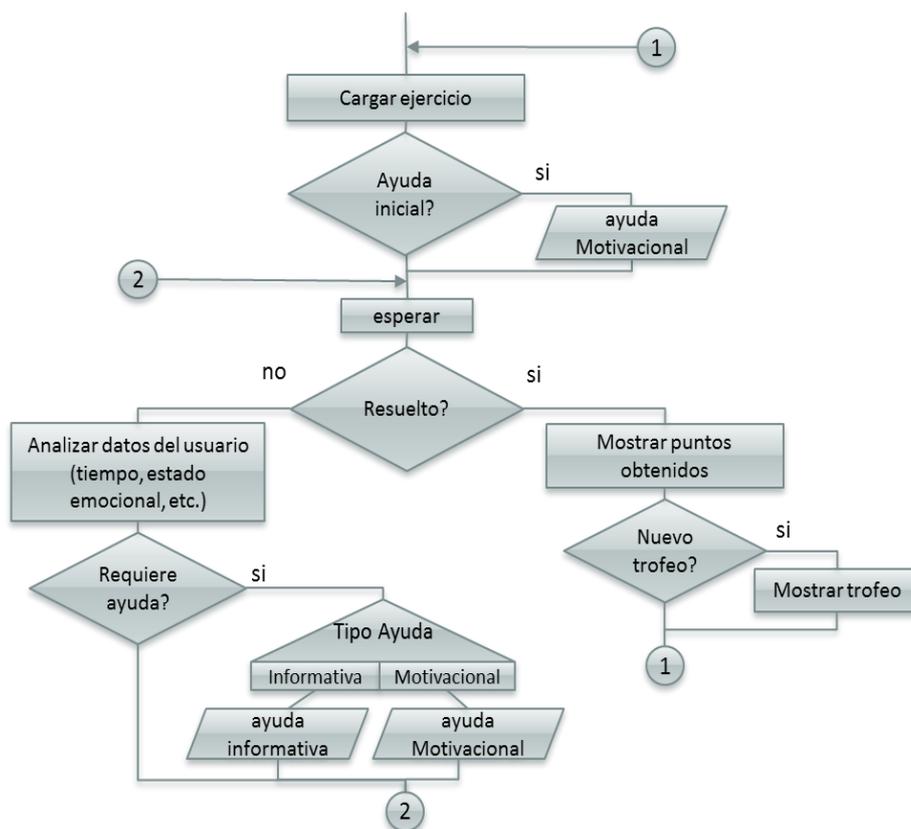


Fig. 4. Diagrama de flujo que muestra el proceso que siguen los cursos de EasyLogic.

La interfaz principal para resolver los ejercicios se muestra en la Figura 5, la cual consiste en una barra de opciones que re direccionan hacia otras pantallas (1), el nombre del curso que se está realizando (2), el nivel del ejercicio actual (3), la clasificación de los bloques gráficos (4), un área para crear el algoritmo por medio de los bloques (5), el juego que se animará una vez ejecutado el algoritmo (6), la cantidad de bloques

disponibles (7), un botón para ejecutar el algoritmo y otro para ver el código asociado al algoritmo creado (8).



Fig. 5. Interfaz para la solución de ejercicios.

La Figura 6 muestra un ejemplo de una ayuda informativa acerca de la estructura repetitiva Desde/Hasta (For).



Fig. 6. Ejemplo de ayuda Informativa.

En la sección libre se pueden crear algoritmos mediante bloques. Está enfocado principalmente para resolver ejercicios comúnmente utilizados por maestros en las aulas de clase. Una vez elaborado el algoritmo, los estudiantes pueden visualizar el código asociado en JavaScript. En la Figura 7 se muestra un ejemplo de un algoritmo,

donde al ejecutarlo, se tendrá que introducir un número por teclado y el programa mostrara un mensaje de texto que indicará si eres mayor o menor de edad.

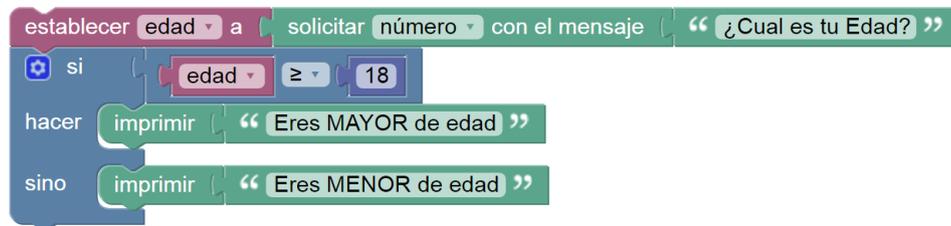


Fig. 7 - Ejemplo de algoritmo creado en EasyLogic.

5. Experimentos y resultados

Hasta la fecha hemos realizado una serie de pruebas preliminares con 10 estudiantes del Instituto Tecnológico de Culiacán. Los sujetos de prueba realizaron 2 algoritmos en la sección libre y completaron todos los cursos que actualmente están disponibles en EasyLogic (Estructuras: Secuencial, Condicional, Repetitiva While y Repetitiva For). Al finalizar los ejercicios, se encuestaron a los sujetos de prueba utilizando la escala de Likert. La Figura 8 muestra los resultados de dos de las preguntas más importantes de la encuesta (acerca de la usabilidad y de las ayudas de la herramienta):

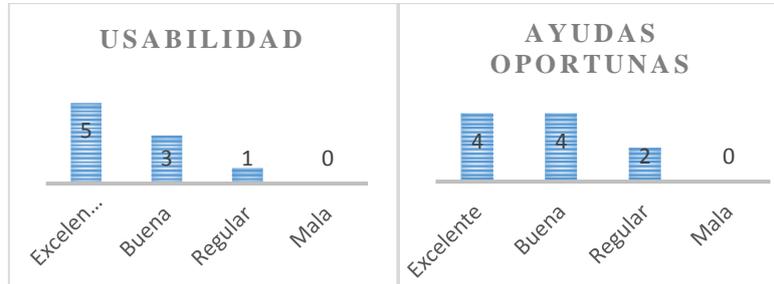


Fig. 8. Resultados de 2 preguntas de las encuestas.

La mayoría de los sujetos de prueba concordaron en que las ayudas mostradas por el sistema han sido oportunas y contribuyen de forma adecuada a la realización de los ejercicios. Otro dato interesante es que los usuarios piensan que el sistema es muy amigable y fácil de usar, incluso sin conocimientos previos del sistema.

Para la realización de los ejercicios, EasyLogic combina 3 aspectos importantes: Afecto, la Herramienta y la Gamificación. La Figura 9 muestra los resultados generados por un estudiante, al momento de realizar el ejercicio 5 del curso “Estructura Selectiva”. La parte Afectiva (A) evaluó el estado emocional del estudiante utilizando PREMOC, al cual se le envía una imagen cada 15 segundos. Se registró que el estudiante sobre todo se encontraba aburrido, por lo que procedió a mostrar una Ayuda Motivacional, debido a que el usuario empleó mucho tiempo para resolver el ejercicio y se había detectado 3 veces la emoción de enganchado, a los 120 segundos se mostró una Ayuda

Informativa. La Herramienta (B), mostró el juego a resolver y proporcionó los bloques adecuados para dar solución al problema. La parte de Gamificación (C), calculó un total de 73 puntos de un máximo de 100, debido a que el usuario tardó bastante tiempo (168 segundos) y además requirió de 2 ejecuciones del algoritmo. Para este ejercicio no se obtuvo ningún trofeo nuevo, debido a que necesitaba 500 puntos para obtener el siguiente trofeo. Opcionalmente el estudiante tenía acceso a la tabla de posición donde se muestra el puntaje actual de todos los integrantes del experimento.

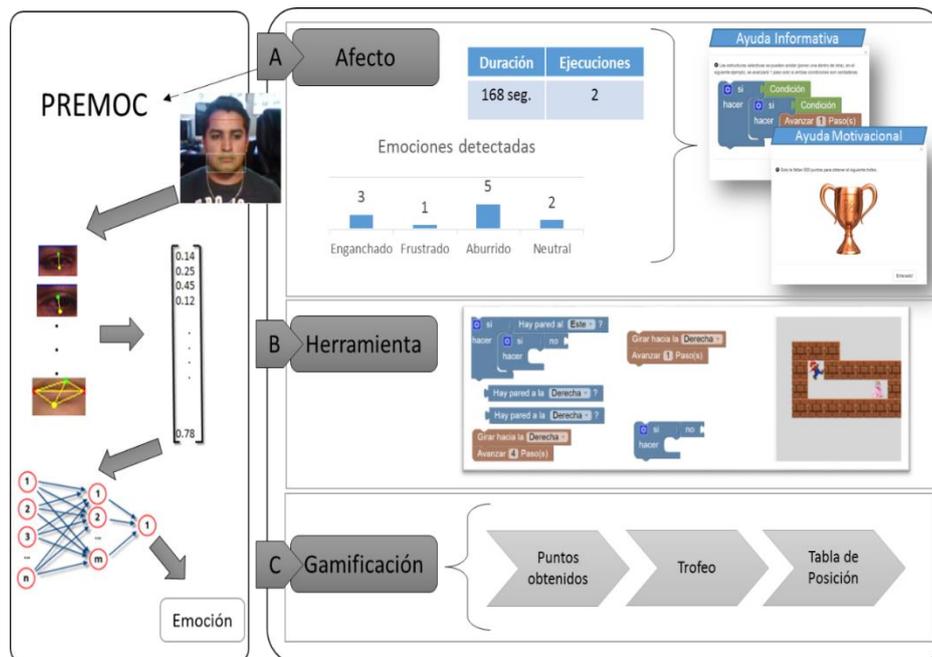


Fig. 9. Ejemplo de experimento Ejercicio 5 del curso "Estructura Selectiva".

6. Conclusiones y trabajos futuros

Aprender a programar sigue siendo una tarea demandante para los estudiantes. En este artículo, presentamos EasyLogic, un STI Afectivo para el aprendizaje de lógica algorítmica y programación. Las pruebas preliminares realizadas, han demostrado que la detección del estado emocional del estudiante, influye favorablemente al ser utilizadas para realizar una intervención, en caso de que el estudiante lo necesite.

Como trabajo futuro se planea realizar experimentos con dos grupos de estudiantes que actualmente cursan la materia de Algoritmos y Lenguajes de Programación, en el Instituto Tecnológico de Culiacán. Se planea modificar la herramienta para desactivar el reconocimiento emocional y experimentar con esta versión, con uno de los grupos de prueba. De este modo los estudiantes no recibirán ayudas informativas, ni motivacionales por parte de la herramienta, solo se mostrarán ayudas iniciales. De tal manera, se permitirá corroborar si la parte afectiva del tutor, implica realmente una mejora en el aprendizaje de los estudiantes.

Referencias

1. Ioannidou, A., Bennett, V., Repenning, A., Koh, K.H., Basawapatna, A.: Computational Thinking Patterns. Online Submiss, Vol. 2 (2011)
2. Wing, J.: Computational thinking. J. Comput. Sci. Coll., Vol. 24, No. 6, pp. 6–7 (2011)
3. Graesser, A.C., Conley, M.W., Olney, A.: Intelligent tutoring systems. APA educational psychology handbook. Vol 3: Application to learning and teaching., Washington: American Psychological Association, pp. 451–473 (2012)
4. Charsky, D.: From Edutainment to Serious Games: A Change in the Use of Game Characteristics. Games Cult., Vol. 5, No. 2, pp. 177–198, Apr. (2010)
5. Seaborn, K., Fels, D.I.: Gamification in theory and action: A survey. Int. J. Hum. Comput. Stud., Vol. 74, pp. 14–31 (2014)
6. Kölling, M.: The Greenfoot Programming Environment. Vol. 10, No. 4, pp. 1–21 (2010)
7. Werner, L., Campe, S., Denner, J.: Children Learning Computer Science Concepts via Alice Game-programming. Proc. 43rd ACM Tech. Symp. Comput. Sci. Educ., pp. 427–432 (2012)
8. Crawford, S.C.P., Dominguez, J.J.D.V.: MIT App Inventor: Enabling Personal Mobile Computing, p. 3 (2013)
9. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J.Y., Silverman, B., Kafai, Y.: Scratch: Programming for All. Commun. ACM, Vol. 52, pp. 60–67 (2009)
10. Brennan, K., Resnick, M.: New frameworks for studying and assessing the development of computational thinking. Annu. Am. Educ. Res. Assoc. Meet. Vancouver, BC, Canada, pp. 1–25 (2012)
11. Repenning, A., Webb, D.: Scalable Game Design and the Development of a Checklist for Getting Computational Thinking into Public Schools (2010)
12. Code Studio.: <https://studio.code.org>. Accesed April (2016)
13. Loyarte, H., Novara, P.: Desarrollo e implementación de un Intérprete de Pseudocódigo para la Enseñanza de Algorítmica Computacional. (2006)
14. D’Mello, S., Olney, A., Williams, C., Hays, P.: Gaze tutor: A gaze-reactive intelligent tutoring system. Int. J. Hum. Comput. Stud., Vol. 70, No. 5, pp. 377–398 (2012)
15. Landowska, A.: Affective computing and affective learning – methods , tools and prospects. Vol. 1, No. 1, pp. 16–31 (2013)
16. D’Mello, S., Graesser, A.: AutoTutor and Affective AutoTutor: Learning by Talking with Cognitively and Emotionally Intelligent Computers that Talk Back. Vol. 1, No. 212, p. 30 (2012)
17. Kazimoglu, C., Kiernan, M., Bacon, L., MacKinnon, L.: A serious game for developing computational thinking and learning introductory computer programming. Procedia - Soc. Behav. Sci., Vol. 47, pp. 1991–1999 (2012)
18. Tessler, J., Beth, B., Lin, C.: Using cargo-bot to provide contextualized learning of recursion. Proc. ninth Annu. Int. ACM Conf. Int. Comput. Educ. Res. - ICER ’13, p. 161 (2013)
19. Butler-Kisber, L.: Mind, Brain, and Education: Implications for Educators. Learn. Landscapes, Vol. 5, No. 1, pp. 1–266 (2011)
20. Utting, I., Cooper, S., Kölling, M.: Alice, greenfoot, and scratch--a discussion. ACM Trans., Vol. 10, No. 4, pp. 1–11 (2010)
21. Fraser, N.: Google Blockly-a visual programming editor. URL: <http://code.google.com/p/blockly>. Accesed Aug (2014)
22. Langner, O., Dotsch, R., Bijlstra, G.: Presentation and validation of the Radboud Faces Database. Cogn. (2010)

Evaluando afecto en un entorno de aprendizaje para Java

Ramón Zatarain-Cabada¹, María Lucía Barrón-Estrada¹,
Francisco González-Hernández¹, Giner Alor-Hernandez²

¹ Instituto Tecnológico de Culiacán, Culiacán, Sinaloa,
México

² Instituto Tecnológico de Orizaba, Orizaba, Veracruz,
México

rzatarain@itculiacan.edu.mx, galor@itorizaba.edu.mx

Resumen. Los cursos de programación habitualmente tienen los índices más altos de abandono. En este artículo se muestra que además de los factores cognitivos como el razonamiento y la memorización, el factor motivacional juega un papel crucial para el aprendizaje del primer lenguaje de programación. También, presentamos el desarrollo y evaluación de un entorno de aprendizaje afectivo e inteligente para Java. El entorno fue probado con estudiantes de universidad y preparatoria donde el reconocimiento afectivo fue usado solamente con los estudiantes de universidad. Se presentan los resultados y las discusiones de las pruebas.

Palabras claves: Inteligencia artificial, entorno inteligente de aprendizaje, entorno colaborativo, detección de afecto multimodal.

Evaluating Affect in a Learning Environment for Java

Abstract. Programming courses usually have the highest dropout indices. In this paper we show that, as cognitive factors such as reasoning and memorization play a crucial role for learning of a first programming language, motivation is also essential in this process. Also, we present the developing and evaluation of an affective and intelligent learning environment for Java. The environment was tested with different university and high-school students where the affective recognition was used only with university students. We present results and discussions about the tests.

Keywords: Artificial intelligence, intelligent learning environment, collaborative environment; affective multimodal detection.

1. Introducción

Históricamente, aprender un lenguaje de programación suele ser un reto complejo para los estudiantes sin experiencia en el área de las ciencias computacionales, debido a que ellos necesitan entender nuevos conceptos de programación como estatutos, expresiones, estructuras de datos y estructuras de control. Además, los estudiantes necesitan desarrollar nuevas habilidades como la resolución de problemas, el desarrollo

de software o la depuración de código. Otros procesos cognitivos se vuelven esenciales para alcanzar el éxito en este reto como la abstracción o el pensamiento matemático. Además, también es importante considerar la obtención de habilidades como aprender las mejores prácticas de programación o el correcto uso de los paradigmas de los lenguajes de programación (orientado a objetos por ejemplo) [1-3].

Anteriormente, se presentó un Entorno Inteligente de Aprendizaje (EIA) llamado Java Sensei [4], el cual detecta el afecto del estudiante a través de sus características faciales. Además, Java Sensei utiliza estrategias pedagógicas utilizando el afecto detectado. En este trabajo, se presentan dos características nuevas añadidas al EIA. La primera característica agregada es la colaboración social entre los estudiantes que lo utilizan, para esto se analizó la literatura del área conocida como “*Computer-supported Collaborative Learning (CSCL)*” [5].

La segunda característica agregada es el reconocimiento de afecto multimodal, el cual detecta las emociones de los estudiantes a partir de diferentes dispositivos que se encuentran interactuando con ellos. De este modo, se mejora la precisión y exactitud para detectar el estado afectivo del estudiante [6].

Este trabajo está estructurada de la siguiente manera: La sección 2 muestra los trabajos relacionados en CSCL y reconocimiento de afecto multimodal; La sección 3 describe los cambios realizados en Java Sensei; La sección 4 describe los experimentos realizados con el EIA y sus resultados. La sección 5 presenta las discusiones de los resultados, conclusiones y el trabajo a futuro.

2. Trabajos relacionados

2.1. CSCL en los EIA

Un ejemplo de los efectos de la colaboración social es dada en [7] donde los estudiantes usan un chat después de la lectura de material académico y la realización de un examen sobre sus conocimientos. El contenido del texto generado en el chat fue analizado y en él encontraron que los estudiantes compartieron información por medio de sus opiniones y debates. El trabajo reportado en [8] analiza las actividades que los estudiantes llevan a cabo durante la colaboración y las relaciones entre esas actividades, todo mientras utilizan un software en línea. El software fue utilizado en un salón de clases donde los estudiantes podrían hablar cara a cara, además, los maestros podían entrar al software para responder las preguntas del estudiante. Los resultados probaron que la colaboración ayudo a los estudiantes a realizar tareas más complejas y de larga duración aplicando métodos de planeación y estrategia decididas en conjunto. Järvelä [9] analizo varios diseños de herramientas CSCL para encontrar características comunes entre ellas. El análisis identifico tres diseños principales: 1) Incrementar la conciencia acerca del conocimiento propio, 2) Soportar la expresión de ideas en el proceso de aprendizaje de los estudiantes, 3) Monitorear la adquisición y activación de procesos regulatorios.

2.2. Reconocimiento de afecto multimodal

El Sistema Java Tutor [10] es un sistema tutor afectivo el cual usa diálogos en lenguaje natural aplicándolo en una interfaz humano a computadora o humano a

humano. El sistema aplica la teoría cognitiva ACT-R para la representación del conocimiento. Diferentes sensores en el tutor miden las señales eléctricas del cerebro, además, una cámara detecta las expresiones faciales. En ABE [15] se presenta un framework basado en agentes. Los agentes son componentes que administran el procesamiento y el flujo de la información obtenida desde diferentes dispositivos los cuales funcionan a un bajo nivel. ABE busca una solución estándar para configurar y agregar diferentes dispositivos detectores de afecto en cualquier software. FILTWAM [11] es otro framework que realiza reconocimiento de afecto en tiempo real y no intrusivo dentro de un entorno de juegos educativos. El framework está dividido en 4 capas, cada una con un rol y componentes específicos. FILTWAM presenta algunas técnicas para permitirle al usuario clasificar las fotos que le son tomadas.

Uno de las principales ventajas de Java Sensei sobre otros sistemas similares es que este sistema está enfocado en mejorar las buenas prácticas de programación. Por ejemplo, cuando el estudiante encuentra una solución a un problema, el sistema recomienda una solución mejor u óptima. Otra ventaja de Java Sensei es su sistema de recomendación que usa un modelo del estudiante para personalizar la selección de ejercicios y las ayudas y consejos dependiendo de las preferencias del estudiante. Respecto al lado tecnológico, la principal ventaja de Java Sensei es que puede ser usado en cualquier tipo de computadora (PC, laptop, móvil) que contenga un navegador web y tenga instalado una cámara y un micrófono, lo cual es común hoy en día. A diferencia de otros sistemas similares, Java Sensei no necesita sensores especiales para el reconocimiento de emociones.

3. Integración de características en Java Sensei

3.1. Colaboración social

A continuación, se describirán los nuevos elementos añadidos a Java Sensei. Estos elementos no tratan de convertir al EIA en un entorno CSCL. Sin embargo, se considera que las características colaborativas ayudaran al EIA a tener más herramientas para ayudar a los estudiantes en sus procesos de aprendizajes. La literatura sobre CSCL se resumió en cuatro componentes sociales: sala de chat, debates, consejos y roles sociales.

Sala de Chat: Se agregó una sala de chat donde los estudiantes pueden hablar entre ellos mientras realizan un ejercicio. Cuando un estudiante inicia un ejercicio, el sistema lo asigna dentro de un equipo; posteriormente, el sistema elige un líder basándose en las habilidades globales de cada estudiante. Todos los miembros necesitan trabajar al mismo tiempo.

Debates: Java Sensei ya usaba un enfoque de resolución de problemas el cual es usado típicamente en los entornos CSCL también. Los entornos CSCL tratan de persuadir a los estudiantes de realizar discusiones y debates sobre el tema que están resolviendo para así llegar a una solución en común. Basándose en eso, se realizó un cambio en la parte donde se resolvían problemas en el EIA. Cuando se elige una respuesta, todos los miembros deben elegir la misma respuesta para poder continuar, en caso de que no lleguen a un acuerdo, los miembros deberán ejercer los roles sociales que les fueron asignados.

Consejos: El EIA trabaja como un miembro consejero dentro el equipo. Propone consejos acorde a los estados afectivos y a los estados cognitivos de los miembros del equipo. Los consejos para los estudiantes están almacenados dentro del dominio del tutor.

Social Roles: Cuando un ejercicio es iniciado en Java Sensei, el sistema asigna a dos miembros del equipo dos roles:

1. Líder, el líder trata de llegar a un acuerdo entre todos los miembros del equipo cuando se presenta un debate. En caso de que no haya acuerdo, él decide la respuesta final.
2. Comunicador, este miembro puede establecer contactos con otros equipos para solicitar ayudas o consejos.

A continuación, se describe el proceso que siguen los estudiantes para realizar un ejercicio dentro de Java Sensei. Cada punto del proceso es mostrado en la Fig 1.

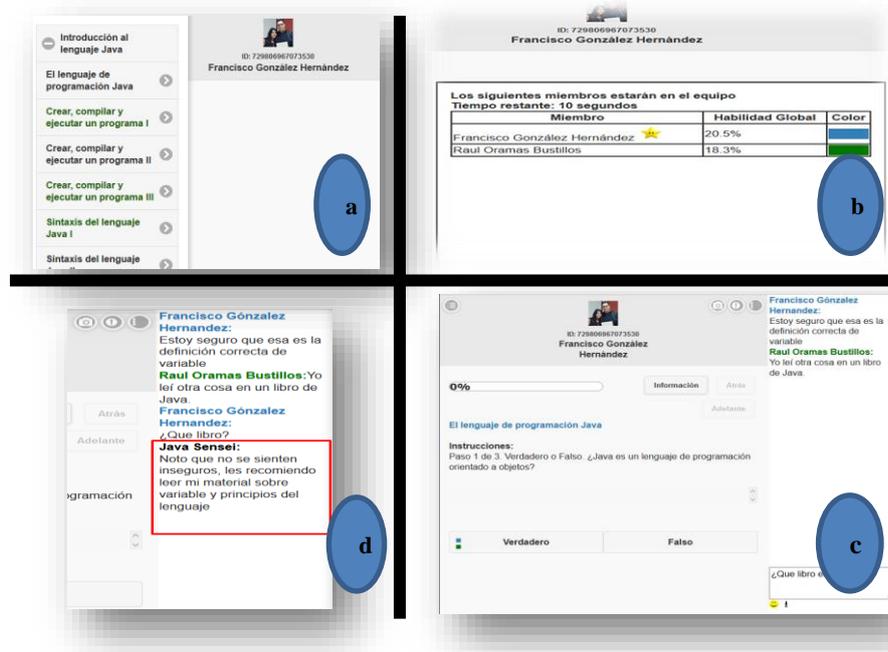


Fig. 1. Colaboración dentro de Java Sensei.

- (a) El estudiante escoge un ejercicio en el menú.
- (b) El sistema trata de emparejar al estudiante con otros estudiantes para poder crear un equipo conformado de dos a cinco miembros. El tiempo de espera para encontrar estudiantes con un conocimiento similar es de 30 segundos.
- (c) Durante la realización de los ejercicios, el tutor del sistema invita a los estudiantes a que se comuniquen entre ellos. En el momento que los estudiantes escogen una respuesta, se les da la posibilidad de compartir mensajes de voz y de texto.

- (d) En caso de que el sistema decida realizar una intervención, se mostrará como un mensaje la sala de chat.

3.2. Detección de afecto a través de diálogos

Java Sensei implementa un algoritmo semántico (ASEM) el cual es utiliza etiquetas semánticas para identificar la emoción del estudiante a partir de sus diálogos de texto [12], el algoritmo trabaja de la siguiente manera:

- (a) Un estudiante ingresa un texto en la sala de chat.
- (b) El texto es normalizado, esto quiere decir que a las palabras se les remueven los acentos, números y caracteres especiales. Además, las letras en mayúsculas son convertidas a minúsculas.
- (c) Las palabras que no representan una emoción como “*ella, el, eso*” son removidas por otro corpus nombrado como *StopWords*.
- (d) Las palabras semánticas son buscadas en dos corpus uno *Semántico* y el otro llamado *Palabras Impropias*. Si la palabra semántica no es encontrada en alguno de los corpus entonces es agregada a un corpus de *Nuevas Palabras*.
- (e) Si la palabra semántica es encontrada en el corpus, entonces dos características de la palabra son extraídas, el *Probability Factor of Affect* (PFA) y la emoción.
- (f) La emoción es clasificada acorde a las características obtenidas de la palabra y se utiliza un algoritmo potenciador/inhibidor (explicado más adelante).
- (g) La emoción con la intensidad más alta es producida.

El clasificador ejecuta un algoritmo potenciador/inhibidor para el valor encontrado en cada palabra, el proceso es el siguiente:

- (a) La palabra semántica es obtenida.
- (b) Esta palabra es buscada en el corpus *Semántico*. Si es encontrada, la emoción y el valor PFA son extraídos.
- (c) Si la palabra es un adverbio negativo, entonces la emoción de la siguiente palabra es invertido (Por ejemplo, de feliz a triste).
- (d) Si es la palabra es un adverbio superlativo, se identifica si la palabra es un pre-adverbio o un post-adverbio. En caso de ser un pre-adverbio se incrementa el PFA de la siguiente palabra semántica. En caso de ser un post-adverbio se incrementa el valor PFA de la palabra previa.
- (e) Finalmente, se suma el valor PFA en cada emoción encontrada en el texto de entrada y se determina que la emoción escogida es aquella que tenga el valor PFA más alto.

3.3. Detección de afecto a través de la voz

El algoritmo de clasificación de las emociones utiliza un *Support Vector Machine* (SVM). Para poder obtener las características principales de la voz y para entrenar el SVM, el sistema extrae la intensidad y el ritmo del audio, el cual se obtiene por medio de la sala de chat [13, 14]. Para implementar el SVM se usó la librería Java-VM [15] el cual utiliza el *Nearest Mean Classifier*. Antes de la clasificación, se tuvo que construir un corpus propio de audios. Para lograr esto, se invitó a un grupo de 20 personas (12

hombres y 8 mujeres), de 22 a 30 años de edad a ver una serie de videos, los cuales provocarían las emociones que se buscaban obtener. Después de ver los video se grabó un audio con una duración de al menos de 10 segundos en el cual expresaban sus sentimientos y emociones acerca del video visto. Al último, se les solicito evaluar la prueba self-assessment manikin (SAM) para categorizar la emoción con su correspondiente audio. El algoritmo para extraer el audio y clasificar el afecto de la voz es el siguiente:

- (a) El estudiante ingresa un mensaje de audio por medio de la sala de chat.
- (b) El Sistema extrae del audio un espectro de 1024 *frames*.
- (c) Cada *frame* es procesado obteniendo así el valor más grande y más pequeño de la intensidad y el ritmo.
- (d) Esos cuatro valores son normalizados donde cada valor tiene su propio rango de normalización
- (e) Los cuatros valores son ingresados en el SVM para la clasificación de las emociones.
- (f) El SVM predice la emoción.

3.4. Sistema lógico difuso para el reconocimiento multimodal

Java Sensei usa un conjunto de reglas difusas para la evaluación pedagógica de los estudiantes. Las reglas cuentan con 5 variables de entrada y 3 variables de salida. Las variables de entrada son: 1) emoción del rostro, 2) emoción del habla, 3) emoción del dialogo, 4) habilidad global (en referencia a su dominio sobre el tema) y 5) calidad de las respuestas (en referencia al ejercicio actual). Las variables de salida son: 1) retroalimentación (parte cognitiva), 2) expresión (apoyo afectivo) e 3) intervención (mensajes de parte del agente pedagógica). El EIA ejecuta el siguiente procedimiento cuando los estudiantes realizan sus ejercicios en cualquier navegador web:

- (a) El navegador web del estudiante guarda la información generada por el usuario (imágenes del rostro, mensajes de voz y mensajes de texto).
- (b) Después de un intervalo de 20 segundos, el navegador web envía los datos al servidor. Estos datos están administrados por el sistema.
- (c) El sistema valida que todos los estudiantes envíen sus datos; después de eso, el sistema evalúa cada emoción.
- (d) Si la mayoría de los estudiantes necesitan una intervención, el sistema enviara un mensaje el cual se construye a partir de las variables de salidas de expresión y retroalimentación.
- (e) El sistema mostrará la información en el navegador del estudiante.

La Fig. 2 muestra la interconexión entre las diferentes fuentes de datos afectivos y el procedimiento que realiza el EIA.

4. Experimentos, resultados y discusiones

El experimento de este trabajo se comparó con el experimento reportado previamente en [16] con una versión no multimodal ni colaborativa del sistema, esto con el objetivo de poder comparar si hubo mejoras después de los cambios. El primer

experimento fue llevado a cabo con un grupo de 20 estudiantes universitarios de ingeniería en sistemas del Instituto Tecnológico de Culiacán (México). El tema evaluado fue “herencia” y se consideraron diferentes puntos como el conocimiento sobre herencia y los buenos hábitos de programación. Java Sensei detectaba emociones mientras los estudiantes usaban el entorno. Se realizó una evaluación pre-test y otra post-test. Los resultados del experimento son mostrados en la Fig. 3.

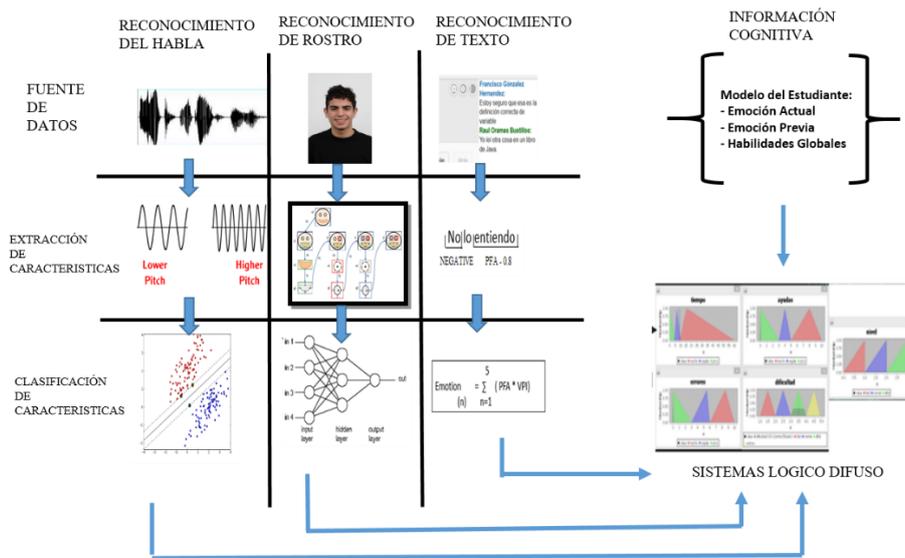


Fig. 2. Unificación entre el reconocimiento multimodal y los aspectos cognitivos.

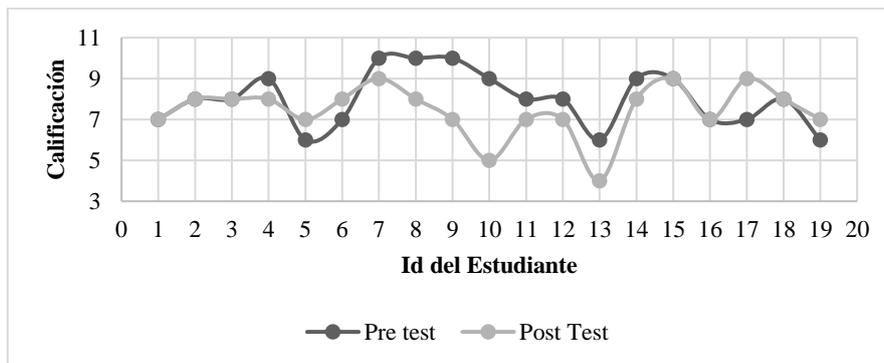


Fig. 3. Resultados con estudiantes de nivel universitario.

En el experimento anterior la muestra fue dividida en dos grupos: control y experimental. Los resultados reportan que los 19 estudiantes del grupo experimental mejoraron un 44.4% sus calificaciones en comparación al grupo control, 35% se mantuvieron sin cambios y 21% bajo sus calificaciones.

El segundo experimento fue llevado a cabo con alumnos de preparatoria en el Instituto Chapultepec (Culiacán, Sinaloa, México). La muestra consistía en treinta y dos estudiantes de dos grupos con una edad promedio de 15.125 años y una desviación estándar de 0.33. Antes de realizar los experimentos, los estudiantes se encontraban tomando un curso introductorio en el lenguaje de programación JavaScript. Se eligieron cuatro temas: Introducción a Java, variables y cálculos, estatutos de selección e estatutos de iteración. Todos esos temas son cubiertos por el dominio de Java Sensei. Se realizaron tres sesiones de 45 minutos, para manejar los mismos tiempos que se utilizaron en el primer experimento. En la primera sesión, los estudiantes resolvieron el pre-test en papel. En la siguiente sesión, los estudiantes usaron el EIA. En la última sesión, los estudiantes resolvieron el post-test en papel. Los maestros tuvieron instrucciones en como poder usar el EIA.

Ambas pruebas contuvieron 15 preguntas de tres tipos: el tipo 1 el cual evalúa conceptos teóricos con opciones “verdadero y falso”; el tipo 2 el cual también evalúa conceptos teóricos, pero con opciones múltiples; el tipo 3 el cual evaluaba la habilidad para entender una pieza de código o un programa completo. La tabla 1 muestra la distribución del tipo de pregunta en ambos exámenes.

Tabla 1. Distribución del tipo de preguntas en ambos exámenes.

Tipo de pregunta	Pre-test	Post-test
Uno	2	2
Dos	6	3
Tres	7	10

La puntuación de la prueba es un número entre 0 y 100. Se describen los análisis estadísticos descriptivos de los resultados de ambas pruebas. La tabla 2 muestra la media, la mediana y sus correspondientes medidas de dispersión la cuales son la desviación estándar (DE), el primer cuartil (PC) y el tercer cuartil (TC).

Tabla 2. Resultados de las pruebas.

Prueba	Media	DE	Mediana	PC	TC
Pre-test	46.7	12.36	46.7	40	53
Post-test	37.29	10.78	40	26.67	40

Además, se evaluó la confiabilidad de ambas pruebas del segundo experimento (pre y post tests). Para evaluar ambas pruebas, se usó la metodología descrita en [17]. La metodología mide la complejidad de las preguntas asignándole una categoría a cada una. La ecuación 1 es usada para obtener el Índice de Confiabilidad (IC):

$$IC = \text{respuestas correctas} / \text{total de preguntas.} \quad (1)$$

El IC es comparado y clasificado en una escala mostrada en la metodología. La escala tiene las categorías: muy fácil, fácil, medio, difícil y muy difícil. Los resultados de la medida son mostrados en la Fig. 4.

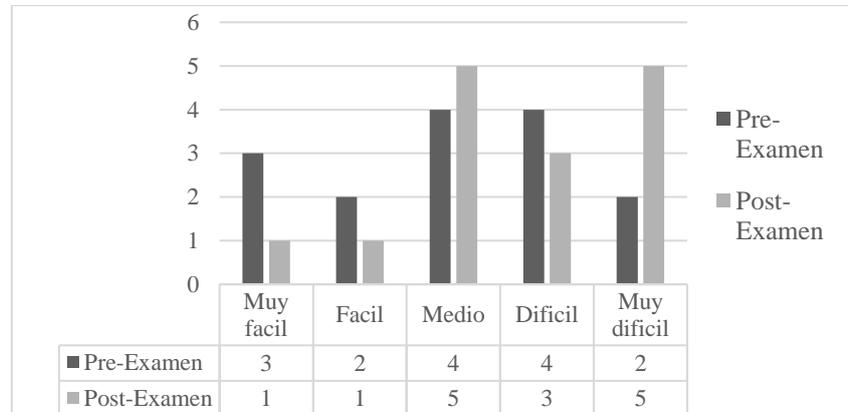


Fig. 4. Comparación del nivel de dificultad en ambas pruebas.

También se midió la media del puntaje obtenido de los cuatro tópicos evaluados en las pruebas. La tabla 3 muestra los resultados.

Tabla 3. Resultados de cada tema.

Tema	Pre-test	Post-test
Introducción a Java	45.57	39.25
Variables y cálculos	29	23
Estatutos de selección	48.25	45.25
Estatutos de iteración	55.33	43.66

Los resultados en la tabla indican que aparentemente los estudiantes en el segundo experimentos no tuvieron un cambio importante en sus resultados, en contraste a los estudiantes de ingeniería del primer experimento mostrados en la Fig. 3. Este resultado está generalizado para todos los estudiantes; la desviación estándar en la Tabla 2 demuestra que el puntaje de los estudiantes casi no excede los 50 puntos de los 100 puntos posibles. Además, los cuartiles muestran que los puntajes son homogéneos. Estos resultados llevan a pensar que las características emocionales tiene un impacto importante en los estudiantes de modo que se estudiaría el impacto de agregar más detectores emocionales. Es importante considerar todas esas características. La importancia de entender la influencia de las emociones en los estudiantes es algo que merece una considerable atención.

Una observación que se averiguo en el análisis de confiabilidad de los exámenes es que los estudiantes tuvieron problemas para entender los temas introductorios. El aprender nuevos conceptos es una tarea siempre complicada. Al contrario, los temas relacionados con razonamiento de programación o de codificación tuvieron pocos problemas. Todo esto se refleja en la Tabla 3. Además, otro importante factor (y conclusión) es que los estudiantes de preparatoria están menos comprometidos para aprender un tema como la programación a comparación de los estudiantes de ingeniería. Esto ayuda a entender que temas necesitan más atención en el diseño de las pruebas con respecto a los estudiantes con un menor nivel académico.

Por otra parte, se observa que el post-test fue diseñado con una complejidad más grande en los temas de “Introducción a Java” y “Variables y cálculos”. Por ejemplo, en la Fig. 4 se observa que ambos exámenes tienen 8 preguntas con un nivel más duro. Esto vuelve necesario la redefinición de las pruebas. Posiblemente, se tenga que modificar la distribución de las preguntas mostradas en la Tabla 1.

5. Conclusiones y trabajo a futuro

En este trabajo se ha aprendido la complejidad de unir los conceptos usados en CSCL con detección multimodal para enseñarles a los estudiantes nuevos conceptos de un lenguaje de programación. Integrar todo esto es un reto que ayudará a entender que aspectos son necesarios para ayudar a los estudiantes. Es complicado considerar que solamente una prueba pueda evaluar todos los tipos de grupos. En el futuro, se realizarán más pruebas para validar la confiabilidad de la integración multimodal. Otra importante meta es trabajar más para mejorar la clasificación y la integración de diferentes reconocedores en Java Sensei. Finalmente, se quiere agregar más técnicas con respecto al aprendizaje colaborativo.

Referencias

1. McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y.B.-D., Laxer, C., Thomas, L., Utting, I., Willusz, T.: A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. *ACM SIGCSE Bulletin*, Vol. 33, No. 4, p. 125–180 (2001)
2. Matthíasdóttir, Á.: How to teach programming languages to novice students? Lecturing or not. *International Conference on Computer Systems and Technologies-CompSysTech*. (2006)
3. Robins, A., Rountree, J., Rountree, N.: Learning and teaching programming: A review and discussion. *Computer Science Education*, Vol. 13, No. 2, pp. 137–172 (2003)
4. Zatarain Cabada, R., Barron Estrada, M.L., Gonzalez Hernández, F.: An Affective Learning Environment for Java. *Advanced Learning Technologies (ICALT), IEEE 15th International Conference on, IEEE* (2015)
5. Dillenbourg, P., Järvelä, S., Fischer, F.: The evolution of research on computer-supported collaborative learning. *Technology-enhanced learning*, Springer, pp. 3–19 (2009)
6. D'mello, S.K., Kory, J.: A Review and Meta-Analysis of Multimodal Affect Detection Systems. *ACM Computing Surveys (CSUR)*, Vol 47, No. 3, p. 43 (2015)
7. Dowell, N.M., Cade, W.L., Tausczik, Y., Pennebaker, J., Graesser, A.: What works: Creating adaptive and intelligent systems for collaborative learning support. *Intelligent Tutoring Systems*, Springer (2014)
8. Janssen, J., Erkens, J., Kirschner, P., Kanselaar, G.: Task-related and social regulation during online collaborative learning. *Metacognition and Learning*, Vol. 7, No. 1, pp. 25–43 (2012)
9. Järvelä, S., Kirschner, P., Panadero, E., Malmberg, J., Phielix, C., Jaspers, J., Koivuniemi, M., Järvenoja, H.: Enhancing socially shared regulation in collaborative learning groups: designing for CSCL regulation tools. *Educational Technology Research and Development*, Vol. 63, No. 1, pp. 125–142 (2014)
10. Wiggins, J.B., Boyer, K.E., Baikadi, A., Ezen-Can, A., Grafsgaard, J.F., Young Ha, E., Lester, J.C., Mitchel, C.M., Wiebe, E.N.: JavaTutor: An Intelligent Tutoring System that

- Adapts to Cognitive and Affective States during Computer Programming. Proceedings of the 46th ACM Technical Symposium on Computer Science Education, ACM (2015)
11. Bahreini, K., Nadolski, R., Westera, W.: Towards multimodal emotion recognition in e-learning environments. *Interactive Learning Environments*, (ahead-of-print), pp. 1–16 (2014)
 12. Wu, C.-H., Chuang, Z.-J., Lin, Y.-C.: Emotion recognition from text using semantic labels and separable mixture models. *ACM transactions on Asian language information processing (TALIP)*, Vol. 5, No. 2, pp. 165–183 (2006)
 13. Krüger, S.E.: Speech recognition with support vector machines in a hybrid system. *INTERSPEECH*, Citeseer, (2005)
 14. Pan, Y., Shen, P., Shen, L.: Speech emotion recognition using support vector machine. *International Journal of Smart Home*, Vol. 6, No. 2, pp. 101–107 (2012)
 15. Abeel, T., Van de Peer, Y., Saeys, Y.: Java-ML: A machine learning library. *The Journal of Machine Learning Research*, Vol. 10, pp. 931–934 (2009)
 16. Barrón-Estrada, M.L., Zatarain-Cabada, R., González Hernández, F., Oramas Bustillos, R., Reyes-García, C.A.: An Affective and Cognitive Tutoring System for Learning Programming. *Advances in Artificial Intelligence and Its Applications*, Springer, pp. 171–182 (2015)
 17. Escudero, E.B., Reyna, N.L., Morales, M.R.: The level of difficulty and discrimination power of the Basic Knowledge and Skills Examination (EXHCOBA). *Revista Electrónica de Investigación Educativa*, Vol. 2, No. 1, p. 2 (2002)

Sistema de reconocimiento multimodal de emociones relacionadas al aprendizaje en dispositivos móviles

María Lucía Barrón-Estrada, Ramón Zatarain-Cabada,
Claudia Guadalupe Aispuro-Gallegos

Instituto Tecnológico de Culiacán, Culiacán, Sinaloa,
México

{lbarron, rzatarain, m03171007}@itculiacan.edu.mx

Resumen. Gran variedad de sistemas reconocedores de emociones han sido implementados, pero pocos han logrado aplicarse en el mundo real debido al elevado costo de la tecnología necesaria y al bajo porcentaje de efectividad del reconocimiento, cuando no se trabaja con emociones espontáneas. Este artículo presenta la implementación de un sistema de reconocimiento multimodal de emociones usando dispositivos móviles y la creación de una base de datos afectiva por medio de una aplicación móvil. El reconocedor puede ser integrado fácilmente a una aplicación educativa móvil para identificar las emociones de un usuario mientras éste interactúa con el dispositivo. Las emociones que el sistema reconoce son compromiso y aburrimiento. La base de datos afectiva fue creada con emociones espontáneas de estudiantes que interactuaron con una aplicación móvil educativa llamada Duolingo y una aplicación móvil recolectora de información llamada EmoData. El sistema desarrollado tiene un porcentaje de efectividad aceptable y a medida que se incrementa el número de registros de la base de datos afectiva, se espera que este porcentaje mejore.

Palabras clave: Computación afectiva móvil, reconocimiento de emociones, dispositivos móviles, reconocimiento voz, sensores de dispositivos móviles.

Multimodal Recognition System of Learning Emotions in Mobile Devices

Abstract. Many emotion recognizer systems have been developed, but only a few of them have been used in the real world. This may occur due to several reasons like: the high cost of required technology and the low rate of accuracy of recognition, when not working with spontaneous emotions. This paper presents the implementation of a multimodal emotion recognition system using mobile devices and the creation of an affective database populated by data collected from a mobile application. The recognizer can be easily integrated into a mobile educational application to identify user's emotions as they interact with the device. The recognized emotions are engagement and boredom. The affective database was created with spontaneous emotions of students who used a mobile educational application called Duolingo and a data gathering mobile application called EmoData. The developed system has an acceptable accuracy rate and by

increasing the amount of records of the affective database, it is expected this rate will be better.

Keywords: Mobile affective computing, emotion recognition, mobile devices, speech recognition, sensors of mobile devices.

1. Introducción

En los últimos años, el avance en el reconocimiento de emociones ha sido impresionante, generándose una gran variedad de sistemas reconocedores [1][2][3], bases de datos afectivas [4], técnicas de extracción de características y clasificación que logran mejorar ciertos aspectos como incrementar la precisión de reconocimiento y disminuir el tiempo de procesamiento, entre otros.

También se ha logrado confirmar la importancia del reconocimiento de emociones en el aprendizaje, definiéndolo como un componente clave en el desarrollo de aplicaciones, tutores y ambientes de aprendizaje inteligentes capaces de responder a las necesidades afectivas del estudiante. Cuando un estudiante está aprendiendo algo nuevo o aplicando sus conocimientos para resolver algún ejercicio pasa por varios estados emocionales, que pueden ser confusión, frustración, aburrimiento, compromiso u otros estados afectivos [5], [6], [7].

A pesar de estos avances, en esos sistemas persisten algunos problemas como: requieren de sensores o dispositivos invasivos y/o costosos, presentan dificultades asociadas con la recolección de datos de forma adecuada y realista para los modelos de aprendizaje máquina, poseen problemas pertenecientes a la generalización a través de contextos, tiempo y diferencias individuales [8], además existe una gran necesidad de recolectar emociones espontáneas a gran escala y una alternativa a este problema puede ser la captación a través de dispositivos móviles.

En la última década, el número de dispositivos móviles creció considerablemente al igual que las aplicaciones disponibles para ellos. El uso de las apps móviles aumentó un 58% durante el 2015 y el tiempo que las personas dedican a los dispositivos móviles aumentó un 117% durante ese mismo año [9]. Además, creció el uso de todo tipo de apps (a excepción de los juegos, que tuvieron un retroceso del 1%). Desde 2013, estos dispositivos ya había logrado conquistar a la sociedad pasando a formar parte de su vida diaria, las personas usan su teléfono inteligente en todo momento [3], el 40% lo revisa cada 10 minutos, 3 de cada 4 personas revisan su teléfono inteligente al despertar, en periodos vacacionales 8 de cada 10 personas lo utiliza por largos periodos de tiempo y de forma muy frecuente.

Las personas pueden realizar numerosas actividades por medio de sus dispositivos móviles, debido a la gran cantidad de aplicaciones disponibles, sus capacidades de detección, procesamiento y comunicación y variedad de sus componentes: acelerómetro, giroscopio, sensor de luz, sensor de proximidad, cámara fotográfica, micrófono, entre otros.

Gran número de personas con diferentes características (edad, sexo, género, etc.) utilizan frecuentemente dispositivos móviles en cualquier lugar u hora el día, para realizar distintas actividades, esto genera grandes volúmenes de información que puede ser usada, entre otras cosas, para captar y reconocer emociones espontáneas de las personas en un ambiente natural. Por otro lado, la capacidad de procesamiento y

componentes de estos dispositivos, puede llevar a cabo la tarea del reconocimiento en tiempo real en cualquier momento y lugar.

La principal contribución de este trabajo es la implementación de un sistema reconocedor de emociones usando dispositivos móviles, el cual puede ser integrado fácilmente a una aplicación móvil educativa, para incorporar la capacidad de identificar el estado afectivo del estudiante y realizar acciones orientadas a mejorar su aprendizaje. Como parte de la contribución del trabajo se generó una base de datos afectiva mediante EmoData, la aplicación móvil desarrollada. Nuestro trabajo es innovador pues está enfocado a detectar emociones relacionadas con el aprendizaje a través del uso natural de los dispositivos móviles, por lo cual su campo de aplicaciones es vasto.

El contenido del artículo se detalla en el siguiente orden: trabajos relacionados con el reconocimiento de emociones son presentados en la Sección 2; la implementación de la aplicación EmoData y su funcionamiento se encuentran en la Sección 3; el experimento para la creación de una base de datos afectiva se muestra en la Sección 4; la implementación del sistema de reconocimiento de emociones se presenta en la Sección 5 y finalmente en la Sección 6 se exponen las conclusiones y trabajo futuro.

2. Trabajos relacionados

En esta sección se presentan algunos importantes trabajos de investigación en el campo del reconocimiento de emociones en dispositivos móviles.

El sistema de reconocimiento móvil presentado por Wu et al. [10] capta imágenes del rostro y realiza el reconocimiento facial con base en distancias entre rasgos faciales y capta las señales de voz y las procesa con base a Coeficientes Cepstrales en las Frecuencias de Mel (MFCCs), Delta MFCC y Delta-Delta MFCC. Este sistema utiliza una máquina de soporte vectorial (SVM por sus siglas en inglés) para clasificar y reconocer las emociones: neutral, felicidad, enojo y tristeza. El porcentaje de efectividad promedio logrado es del 87%.

MoodScope [11] es una aplicación móvil desarrollada para el sistema operativo iOS, que deduce el estado de ánimo del usuario con base al uso de su celular, basado en un modelo estadístico que utiliza información de los SMS, correos electrónicos, llamadas telefónicas, buscador Web y la ubicación del dispositivo para inferir el estado de ánimo del usuario. Esta aplicación es transparente para el usuario ya que se ejecuta en segundo plano, registra la interacción del usuario con su celular y almacena los datos en un servidor a través de una conexión de datos o Wi-Fi.

EmotionSense [12] es una aplicación móvil para detectar la emoción del usuario con base a los hábitos de uso del dispositivo. Esta aplicación registra la ubicación física donde se encuentra el dispositivo y como se está moviendo, el ruido del entorno y además monitorea las llamadas y mensajes, con el objetivo de desarrollar un patrón de hábitos del usuario que le permitan deducir su emoción. Los componentes que utiliza son: GPS, acelerómetro y micrófono.

Samurai fruit [13] es una variación de un juego para dispositivos móviles para iPhone que reconoce la emoción del jugador con base al comportamiento de sus gestos táctiles en la pantalla del dispositivo. Este juego consiste en estrujar y cortar frutas a través de movimientos táctiles; mientras el usuario está jugando, la aplicación captura y registra las coordenadas de cada punto de un golpe, el área de contacto del dedo a

cada punto y la duración del tiempo de un golpe. El área de contacto se utiliza para medir la presión ejercida por los jugadores. Los estados emocionales que identificaron son: Emocionado, Relajado, Frustrado y Aburrido.

3. Implementación de EmoData

EmoData es una aplicación móvil desarrollada para el sistema operativo Android, esta se encarga de monitorear y recolectar información de los componentes del celular para relacionarlos con el estado afectivo y la postura corporal del usuario. Los componentes del celular con los que trabaja son micrófono, acelerómetro y giroscopio.

En la Figura 1 se puede observar la arquitectura de esta aplicación, la cual está formada por las capas de presentación, recolección y datos.

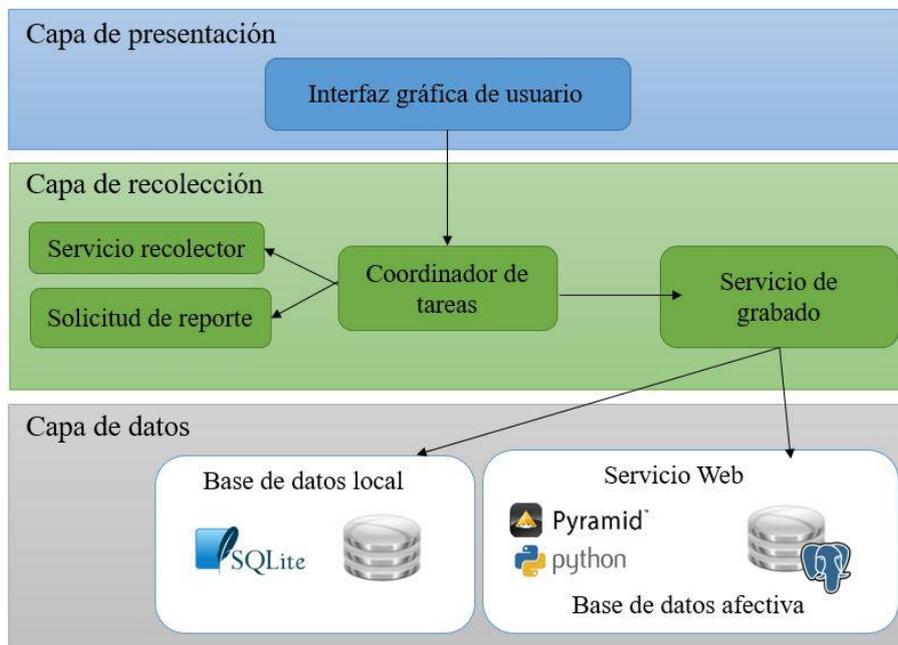


Fig. 1. Arquitectura de EmoData.

Capa de presentación: En esta capa se encuentran las interfaces de EmoData que permiten al usuario interactuar con el sistema para que pueda registrarse, identificarse e iniciar la tarea de recolección. Esta capa utiliza una API de Facebook cuando el usuario desea utilizar su cuenta de Facebook para identificarse.

Capa de recolección: Esta capa se encarga de monitorear y recolectar la información de los componentes del dispositivo y está formada por 2 módulos y 2 servicios explicados a continuación.

- **Coordinador de tareas:** Dirige las tareas principales de la aplicación. Solicita que el servicio recolector obtenga la información de los sensores y después continúa

pidiendo al usuario que ingrese su emoción y postura, por último envía al servicio de grabado la información recolectada, la emoción y postura para que realice el almacenamiento de los datos.

- Servicio recolector: Este servicio se encarga de recolectar el audio recibido por medio del micrófono y la secuencia de valores de los ejes X, Y y Z tanto del acelerómetro como del giroscopio. El audio del micrófono se recolecta en un archivo de audio con formato .pcm y la información del acelerómetro y giroscopio se recolecta como arreglos en formato JSON.
- Solicitud de reporte: Este módulo solicita al usuario que ingrese al sistema su estado emocional y la postura en que se encuentra, con el objetivo de etiquetar la información recolectada automáticamente.
- Servicio de grabado: Este servicio envía al servidor la información recolectada y etiquetada con la emoción y postura del usuario. Se utiliza la librería Volley que hace una solicitud asíncrona HTTP a un servicio Web desarrollado en Python y su Framework Pyramid. El archivo de audio se convierte a formato .wav y se serializa en una cadena base 64 antes para facilitar su envío al servidor, pero la información se almacena localmente en la base de datos SQLite, si no se cuenta con conexión a Internet.

Capa de datos: Esta capa contiene la base de datos afectiva donde se almacena la información, una parte se encuentra en el dispositivo y otra en el servidor.

3.1. Funcionamiento de la aplicación EmoData

En la Figura 2 se esquematiza el funcionamiento de la aplicación EmoData utilizada para la recolección de las señales del dispositivo que se usan para determinar la emoción del usuario.



Fig. 2. Funcionamiento de EmoData.

A continuación se detalla el funcionamiento de EmoData.

1. El usuario inicia la aplicación y el sistema presenta la interfaz inicial del sistema (ver a) Figura 3), donde el usuario puede registrarse o iniciar sesión.
2. Un usuario puede registrarse para iniciar sesión proporcionando sus datos (nombre, número de celular y correo electrónico) y posteriormente ingresa con su usuario y contraseña.

3. El usuario puede iniciar sesión por medio de Facebook o con su usuario y contraseña. Una vez que inicie sesión, la aplicación presenta su interfaz principal mostrada en b) Figura 3.



Fig. 3. Interfaces de EmoData. Las imágenes de la pantalla donde aparecen las posiciones del estudiante fueron tomadas en el artículo de Hosain et al[14].

4. En la interfaz principal, el usuario indica si la tarea de recolección estará activada o no. Durante el proceso de recolección, la aplicación ejecuta tareas recurrentes:
 - Cada 20 segundos se recolectan los valores de los ejes del acelerómetro y giroscopio. El acelerómetro es un dispositivo que detecta las fuerzas de la aceleración y puede medir la aceleración en uno, dos o tres ejes, siendo los más comunes los de tres ejes. Entre los usos del acelerómetro se encuentran detectar el movimiento e inclinación. Por otra parte, el giroscopio es un sensor que permite medir las velocidades angulares de tres ejes, mantener y cambiar la orientación del dispositivo. Un ejemplo de las aplicaciones de estos sensores, es el trabajo de Paredes et al [15] en el cuál se utilizan un acelerómetro y giroscopio para modelar las oscilaciones del brazo durante el uso común del celular. Otra aplicación [16] es utilizar estos sensores para diferencias posturas estáticas (estar de pie, flexionado, sentado o acostado) de posturas dinámicas.
 - Para recolectar la información del acelerómetro y giroscopio la aplicación móvil se comunica con el sistema de sensores y a partir de una instancia de la clase Sensor Manager obtiene acceso a estos sensores para captar los valores de los ejes X, Y y Z en ese momento.
 - El audio recibido por el micrófono se recolecta y almacena en archivos con una duración de 3 minutos por periodo.
 - Cada tres minutos, la aplicación solicita al usuario que reporte su estado afectivo y postura por medio de las interfaces mostradas en c) y d) Figura 3.
 - El almacenamiento de la información se realiza cada cuatro minutos y como se observa en la Figura 4 consiste en enviar la información a un servidor Web en caso de contar con conexión a Internet, si no es posible, entonces se

graba localmente la información para posteriormente volver a intentar enviarla.

- Cuando el usuario decide detener la recolección, la aplicación detiene todas las tareas que esté ejecutando.

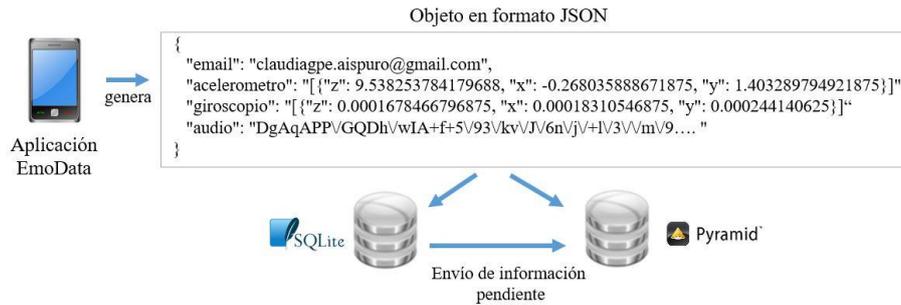


Fig. 4. Grabado de la aplicación EmoData.

4. Método del experimento para la creación de la base de datos afectiva

El objetivo de este experimento es crear una base de datos afectiva de emociones espontáneas que almacene un audio y secuencias de valores de los ejes del acelerómetro y giroscopio etiquetados con la emoción y postura corporal reportadas por el usuario.

El experimento consistió en utilizar la aplicación educativa Duolingo[17] y EmoData para generar la información que se almacenará en la base de datos. Los sujetos de prueba fueron veintiséis estudiantes del Instituto Tecnológico de Culiacán, de los cuales veintiuno son hombres y cinco mujeres de diferentes edades, que van de 18 a 30 años.

La base de datos afectiva fue generada a través de una secuencia de pasos, que se describen a continuación.

Estímulo: Para provocar las emociones se utilizó la aplicación educativa DuoLingo, en la que se realizaron sesiones de práctica de tres minutos. DuoLingo presenta diferentes tipos de ejercicios que pueden ser traducciones de inglés a español y viceversa, ejercicios de pronunciación, escribir la frase que se escucha, ejercicios de opción múltiple, entre otros.

Recolección de señales: Mientras el estudiante practicaba con DuoLingo, las señales de audio e información de los sensores acelerómetro y giroscopio eran registradas por medio de la aplicación EmoData.

Extracción de características: Se desarrolló una aplicación en Java que utiliza la herramienta MusicG[18] para extraer las características de las grabaciones de audio: tono e intensidad y se implementaron los métodos de extracción de características media, desviación estándar de los valores de cada uno de los ejes del acelerómetro y giroscopio y el coeficiente de correlación de los ejes XY.

Clasificación: La clasificación fue realizada automáticamente considerando la posición y emoción reportada por el estudiante.

La base de datos afectiva por el momento está compuesta por 85 registros (entre 3 y 4 por cada estudiante) clasificados con su respectiva emoción y postura corporal.

5. Implementación del sistema de reconocimiento de emociones

El sistema reconocedor de emociones recibe las señales afectivas del usuario y las analiza para retornar como resultado la emoción del usuario. En esta sección se presenta la arquitectura Cliente-Servidor del sistema de reconocimiento. Esta arquitectura se muestra en la Figura 5.

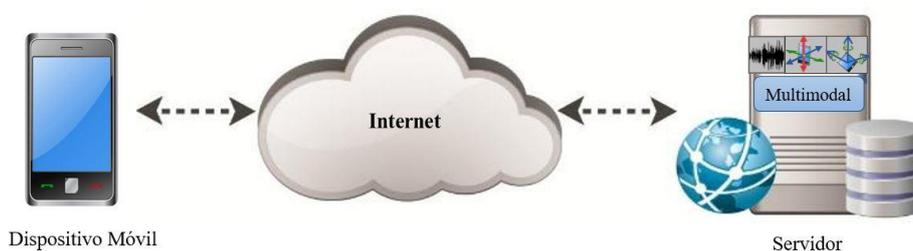


Fig. 5. Arquitectura del Sistema Reconocedor.

El Cliente, en este caso un dispositivo móvil realiza las peticiones o solicitudes de reconocimiento al Servidor, para que realice la tarea de reconocimiento. La arquitectura del servidor está formada por las capas: Servicios Web, Reconocimiento y Datos.

Capa de Servicios Web. La capa de servicios Web está implementada en un servicio Web encargado de recibir las solicitudes de los usuarios y responder con la emoción identificada. En su solicitud, el usuario especifica si desea que se realice reconocimiento unimodal o multimodal y proporciona las señales afectivas para realizar el reconocimiento. Si el reconocimiento es de audio, la señal afectiva es el audio serializado, con la finalidad de facilitar su envío. En el caso del reconocimiento por medio de la postura del usuario, la señal afectiva es la secuencia de valores de los ejes del acelerómetro y por último, en el reconocimiento por medio del movimiento, se utilizan dos señales afectivas, una es la secuencia de valores de los ejes del acelerómetro y la otra una secuencia de valores de los ejes del giroscopio.

Capa de Reconocimiento. La capa de Reconocimiento esta implementada en una librería Java encargada de realizar el proceso de reconocimiento de la emoción con la información que provee la capa de servicio web. Esta librería permite realizar el reconocimiento de forma unimodal, que puede ser por voz, postura o movimiento del usuario o multimodal usando una combinación de los anteriores.

Reconocimiento por voz. Este proceso mostrado en la Figura 6, recibe como entrada un audio en formato wav serializado en Base64 e inicia con la etapa de preprocesamiento, en donde el audio es deserializado y convertido en una onda de audio normalizada a la cual se le extraen dos características: tono e intensidad. La extracción de características se realiza mediante la herramienta MusicG, por medio de esta herramienta se obtienen 4 valores de las dos características: el valor máximo y mínimo del tono y el valor máximo y mínimo de intensidad. Estas características conforman el

vector de características que sirve como entrada a una SVM incluida en la librería LibSVM [19], para obtener la emoción. La máquina de soporte vectorial fue entrenada previamente con la información de la base de datos afectiva.

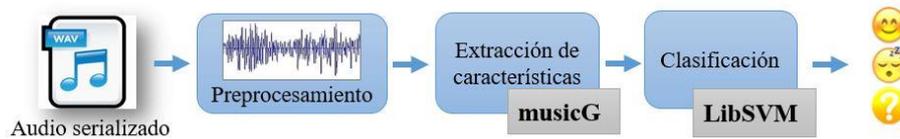


Fig. 6. Algoritmo reconocedor de voz.

Reconocimiento por medio de la posición. Este reconocimiento está basado en el trabajo de Hossain et al. [14], y se muestra en la Figura 7. La entrada es la secuencia de valores de cada uno de los ejes del acelerómetro y en la etapa de preprocesamiento se formatean los datos para facilitar la extracción de sus características.

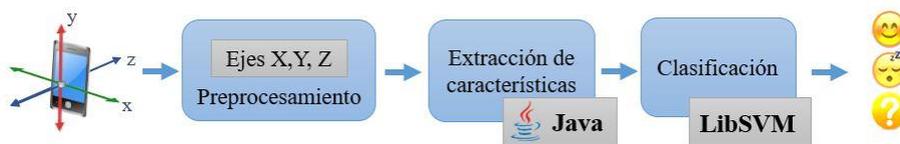


Fig. 7. Reconocimiento por posición.

Para la extracción, se revisaron las técnicas de procesamiento sobre el reconocimiento del contexto a partir de datos del acelerómetro [20] y se seleccionaron algunas características del dominio de tiempo (técnicas matemáticas y estadísticas) que con frecuencia son usadas para extraer información básica de los datos crudos del acelerómetro en algoritmos de reconocimiento de actividad. Entre las características que están disponibles en el dominio de tiempo se encuentran: media, mediana, varianza, desviación estándar, mínimo, máximo, rango, media cuadrática RMS, coeficiente de correlación de señales, entre otros. La media es un promedio de un conjunto de valores o su distribución. Muchas investigaciones han utilizado la media para identificar la postura del usuario (sentado, de pie parado, etc.) [21] y también para diferenciar si la actividad es estática y dinámica [22], entre otros usos.

La varianza es la media de las diferencias con la media elevadas al cuadrado. Y la desviación estándar es la raíz cuadrada de la varianza y representa la variabilidad de un conjunto de datos y la probabilidad de distribución. Los coeficientes de correlación determinan el grado de correlación entre las señales de varios ejes y sirven para medir la fuerza y dirección de una relación lineal entre dos señales. Uno de los más utilizados es el coeficiente de correlación de Pearson. Una de las aplicaciones de la correlación de señales es para diferenciar entre actividades que involucran movimiento en una sola dimensión [23].

Este sistema de reconocimiento extrae la media y la desviación estándar para cada uno de los ejes del acelerómetro y el coeficiente de correlación de Pearson de los ejes XY.

El vector de características está formado por 7 valores: la media y desviación estándar de cada uno de los ejes forman 6 valores y el coeficiente de correlación de Pearson de los ejes XY es el último. Este vector fue creado con un método implementado en Java que extrae las características de las señales. En base a estas características se realiza la clasificación utilizando la SVM de la librería LibSVM previamente entrenada con la información de la base de datos afectiva.

Reconocimiento por medio de movimiento. Este proceso de reconocimiento es llamado “por movimiento” debido a que trabaja con la información de dos sensores de movimiento, tal como se muestra en la Figura 8. Recibe como entrada dos secuencias de valores de los ejes del acelerómetro y del giroscopio. En la etapa de preprocesamiento se formatean los datos para facilitar la extracción de las características. Las características que se extraen son la media y desviación estándar de cada uno de los ejes, por lo que el vector de características está formado por estos 12 valores. Este proceso se realiza con un método implementado en Java. En base a estas características se realiza la clasificación utilizando la máquina de soporte vectorial SVM de la librería LibSVM previamente entrenada con la información de la base de datos afectiva.

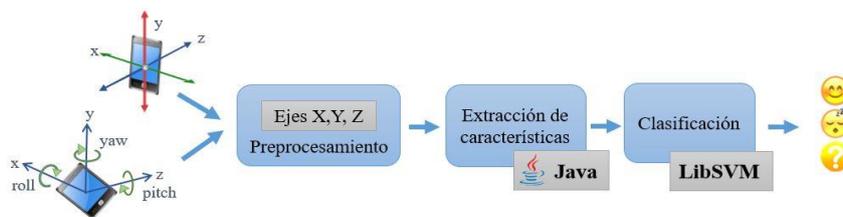


Fig. 8. Reconocimiento por movimiento.

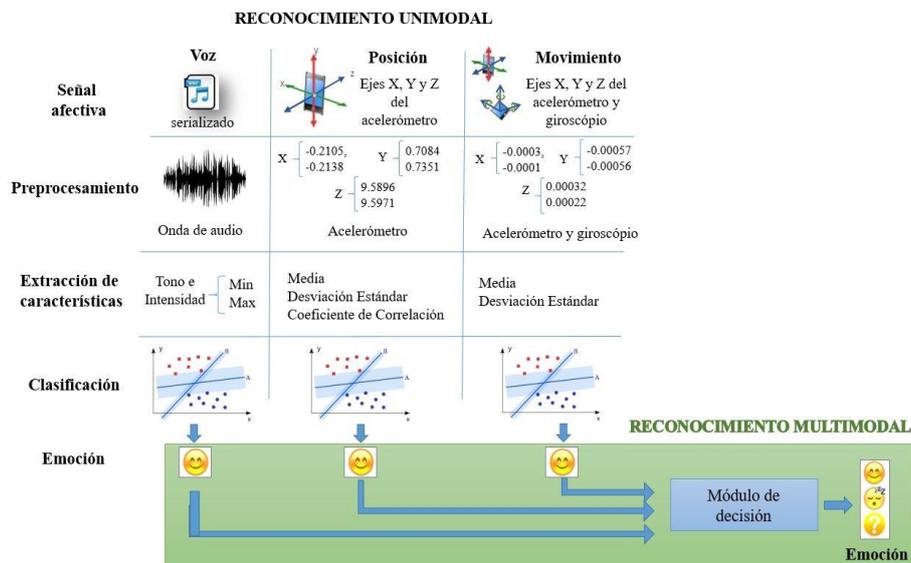


Fig. 9. Proceso de Reconocimiento Multimodal.

Reconocimiento multimodal. Este proceso se representa de forma general en la Figura 9, recibe como entrada las tres señales afectivas: el audio, los datos del acelerómetro y los datos del giroscopio. El método de fusión que se utiliza para combinar las modalidades es a nivel decisión, esto significa que primero se realiza el reconocimiento de emoción de cada una de forma unimodal, identificándose tres emociones, una por cada tipo de reconocimiento realizado y después se fusionan para obtener la emoción final.

El módulo de decisión se encarga de realizar la fusión, este módulo recibe las tres emociones identificadas en los reconocedores unimodales que sirven como entrada a un motor de inferencias difusas por medio de la librería *jFuzzyLogic*[24] que ejecuta un conjunto de reglas difusas, para determinar la emoción final que puede ser Compromiso, Aburrimiento u Otro, este último significa que la emoción no pudo ser clasificada en las otras dos emociones. Un ejemplo de regla difusa se puede observar en la Figura 10.

```
RULE 1: IF eAudio IS aburrimiento AND ePosicion IS compromiso AND
eMovimiento IS compromiso THEN emocion IS compromiso;
```

Fig. 10. Ejemplo de una regla difusa.

Capa de datos. Esta capa se encuentra la base de datos afectiva utilizada para realizar el entrenamiento de la máquina de soporte vectorial SVM.

6. Conclusiones y trabajos futuros

El sistema desarrollado es capaz de realizar el reconocimiento de emociones utilizando la información generada por los sensores de un dispositivo móvil.

El sistema de reconocimiento está implementado de tal manera que puede ser utilizado por diversos tipos de aplicaciones móviles entre ellas, aplicaciones educativas como tutores, juegos o cualquier otra aplicación que necesite incorporar el reconocimiento de emociones de forma continua, no invasiva y en el ambiente natural para el usuario.

El sistema reconocedor está disponible para su uso a través de un servicio web, el cual recibe los datos de entrada (el audio en formato wav serializado, la cadena en formato JSON con la secuencia de los valores de los ejes del acelerómetro y giroscopio) y retorna la emoción identificada como salida. Un ejemplo de la información que el servicio web espera recibir se muestra en la Figura 11.

Secuencia de valores de los Ejes X, Y y Z del acelerómetro en formato JSON.	➔	"[{"eje_z": 9.589645385742188, "eje_x": -0.2105712890625, "eje_y": 0.70843505859375}, {"eje_z": 9.58367919921875, "eje_x": -0.216583251953125, "eje_y": 0.78543589585375}, {"eje_z": 9.595718383789062, "eje_x": -0.213893254958254, "eje_y": 0.73515155489375}]"
Secuencia de valores de los Ejes X, Y y Z del giroscopio en formato JSON.	➔	"[{"eje_z": -0.000213623046875, "eje_x": -0.0003662109375, "eje_y": -0.000579833984375}, {"eje_z": 0.0003204345703125, "eje_x": -0.00018310546875, "eje_y": -0.0005645751953125}, {"eje_z": 0.0002288818359375, "eje_x": 0.00018310546875, "eje_y": 0.00024414062}]"
Audio en formato wav serializado	➔	"DgAqAPPvGQDh/vIA+f+5\93\kv\JV6n\jv+IV3V\mv\9zPAxADgAMAAhAEgAHGA....."

Fig. 11. Ejemplo de los datos de entrada del servicio Web.

Como trabajo futuro se espera realizar nuevas pruebas de sistema de reconocimiento de emociones para determinar la precisión del reconocimiento logrado y analizar los resultados. Así como mejorar el reconocimiento por voz e implementar otras formas de reconocimiento emocional basados en los sensores del celular y agregar la identificación de más emociones relacionadas al aprendizaje.

También se pretende experimentar utilizando otra aplicación educativa para crear la base de datos afectiva y ver cómo influye en la tasa de reconocimiento del sistema.

Referencias

1. Calvo, R.A., D'Mello, S.: Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Trans. Affect. Comput.*, Vol. 1, No. 1, pp. 18–37, (2010)
2. Kory, K.D., Sidney, J.: A Review and Meta-Analysis of Multimodal Affect Detection. Vol. 20, No. 10 (2015)
3. Zhang, S., Hui, P.: A Survey on Mobile Affective Computing. No. 1 (2014)
4. Langner, O., Dotsch, R., Bijlstra, G., Wigboldus, D.H.J., Hawk, S.T., van Knippenberg, A.: Presentation and validation of the Radboud Faces Database. *Cogn. Emot.*, Vol. 24, No. 8, pp. 1377–1388 (2010)
5. Baker, R., Rodrigo, M., Xolocotzin, U.: The dynamics of affective transitions in simulation problem-solving environments. 2nd International Conference on Affective Computing and Intelligent Interaction, pp. 666–677 (2007)
6. D'Mello, S.: A selective meta-analysis on the relative incidence of discrete affective states during learning with technology. *J. Educ. Psychol.*, Vol. 105, No. 4, pp. 1082–1099 (2013)
7. D'Mello, S.K., Graesser, A.: The half-life of cognitive-affective states during complex learning. *Cogn. Emot.*, Vol. 25, No. 7, pp. 1299–308 (2011)
8. Baker, R.S.J.D., D'Mello, S.K., Rodrigo, M.M.T., Graesser, A.C.: Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive-affective states during interactions with three different computer-based learning environments. *Int. J. Hum. Comput. Stud.*, Vol. 68, No. 4, pp. 223–241 (2010)
9. PuroMarketing, <http://www.puromarketing.com/96/26113/uso-apps-moviles-aumento-durante.html>.
10. Wu, Y., Lin, S., Yang, D.: A Mobile Emotion Recognition System Based on Speech Signals and Facial Images. pp. 212–217 (2013)
11. Likamwa, R., Liu, Y., Lane, N.D., Zhong, L.: MoodScope: Building a Mood Sensor from Smartphone Usage Patterns. *MobiSys '13 Proceeding 11th Annu. Int. Conf. Mob. Syst. Appl. Serv.*, pp. 389–402 (2013)
12. Rachuri, K.K., Musolesi, M., Mascolo, C., Rentfrow, P.J., Longworth, C., Aucinas, A.: EmotionSense: A Mobile Phones based Adaptive Platform for Experimental Social Psychology Research. *Int. Stud.*, pp. 281–290 (2010)
13. Gao, Y., Bianchi-Berthouze, N., Meng, H.: What Does Touch Tell Us about Emotions in Touchscreen-Based Gameplay? *ACM Trans. Comput. Interact.*, Vol. 19, No. 4, pp. 1–30, (2012)
14. Bin Hossain, R., Sadat, M., Mahmud, H.: Recognition of human affection in smartphone perspective based on accelerometer and user's sitting position. 17th International Conference on Computer and Information Technology, ICCIT, pp. 87–91 (2015)
15. Paredes, P., Berkeley, U.C., Sun, D.: Sensor-less Sensing for Affective Computing and Stress Management Technology. *Pervasive Computing Technologies for Healthcare and Workshops, Sensor-less Sensing for Affective Computing and Stress Management Technology*, pp. 459–463 (2013)

16. Li, Q., Stankovic, J.A., Hanson, M.A., Barth, A.T., Lach, J., Zhou, G.: Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information. Proceedings 6th International Workshop on Wearable and Implantable Body Sensor Networks, BSN, pp. 138–143 (2009)
17. Duolingo: <https://es.duolingo.com/>
18. MusicG: <https://code.google.com/archive/p/musicg/>
19. Chang, C.-C., Lin, C.-J.: LIBSVM: A Library for Support Vector Machines. ACM Trans. Intell. Syst. Technol., Vol. 2, No. 3 (2011)
20. Figo, D., Diniz, P.C., Ferreira, D.R., Cardoso, J.M.P.: Preprocessing techniques for context recognition from accelerometer data. Pers. Ubiquitous Comput., Vol. 14, No. 7, pp. 645–662 (2010)
21. Jeong, D.U., Kim, S.J., Chung, W.Y.: Classification of posture and movement using a 3-axis accelerometer. International Conference on Convergence Information Technology, ICCIT, pp. 837–844 (2007)
22. Kawahara, Y., Kurasawa, H., Morikawa, H.: Recognizing User Context Using Mobile Handsets with Acceleration Sensors. IEEE International Conference on Portable Information Devices, pp. 1–5 (2007)
23. Ravi, N., Dandekar, N., Mysore, P., Littman, M.: Activity recognition from accelerometer data. Proc. Natl., pp. 1541–1546 (2005)
24. Cingolani, P., Alcalá-Fdez, J.: JFuzzyLogic: A robust and flexible Fuzzy-Logic inference system language implementation. IEEE International Conference on Fuzzy Systems (2012)

Una revisión sistemática de técnicas de gamification en aplicaciones educativas inteligentes

Humberto Marín Vega¹, Giner Alor Hernandez¹,
Ramón Zatarain Cabada², M. Lucía Barrón Estrada²

¹ División de Estudios de Posgrado e Investigación,
Instituto Tecnológico de Orizaba, México

² División de Estudios de Posgrado e Investigación,
Instituto Tecnológico de Culiacán, México

humbert_marin@outlook.com, galor@itorizaba.edu.mx
{rzatarain, lbarron}@itculiacan.edu.mx

Resumen. Gamification es el concepto de la aplicación de la mecánica de juego y técnicas de diseño de juegos para involucrar y motivar a las personas a lograr sus metas. El presente trabajo tiene como objetivo hacer una revisión de las técnicas gamification implementadas en el desarrollo de aplicaciones educativas y la implementación o interacción con otras tecnologías como es el caso de *Serious games*. La importancia de su investigación es debido a que gamification es un término genérico informal para el uso de elementos de juego en sistemas que no son de juego que tienen como objetivo mejorar la experiencia del usuario y la participación de los usuarios. El objetivo de este análisis identificar que atributos de juego y actividades de aprendizaje están presentes en aplicación gamification y juegos serios pero de igual manera están presentes en otras tecnologías como es el caso de ITIL y TEL.

Palabras clave: Atributos del juego, categorías de juego, gamificación, juegos serios.

A Systematic Review of Gamification Techniques in Intelligent Educational Applications

Abstract. Gamification is the concept of applying game mechanics and game design techniques to engage and motivate people to achieve their goals. This paper aims to make a systematic review of gamification techniques implemented in the development of educational applications and the implementation or interaction with other technologies such as serious games. The importance of this research is because gamification is an informal generic term for the use of game elements in systems in order to improve the user experience and user participation. The aim of this analysis is to identify which game attributes and learning activities are present in gamification-based applications, serious games and other technologies such as ITIL and TEL.

Keywords: Game attributes, game categories, gamification, serious games.

1. Introducción

En la educación el aprendizaje basado en el juego es un factor motivador, los juegos en el aprendizaje suelen ser atractivos debido a sus reglas, el sistema de recompensas y sus entornos [1]. La motivación en el aprendizaje de los estudiantes tiene un impacto significativo en el rendimiento y los logros de los estudiantes con el aprendizaje basado en el juego debido a que son mejores que los de la enseñanza tradicional, de esta forma se comprueba que el aprendizaje basado en el juego alcanza el objetivo de aprender de manera efectiva [2]. De acuerdo con este progreso en el aprendizaje basado en el juego, se admiten varios aspectos del proceso de aprendizaje, entre los más importantes está en primer lugar, que se anima a los alumnos a elegir una solución o tomar una decisión en un momento determinado. En segundo lugar, los alumnos pueden comprobar cómo el resultado del juego puede cambiar en función de sus decisiones y acciones. En tercer lugar, se anima a los alumnos a ponerse en contacto con otros miembros del equipo para discutir y negociar los pasos posteriores mejorando entre otras cosas sus habilidades sociales [3]. Los aspectos centrales importantes de los juegos incluyen: búsqueda de información, selección de información, desarrollo de estrategias, resolución de conflictos, ejercicio de la toma de decisiones y la negociación. De acuerdo con estos aspectos, el objetivo y la culminación de los juegos es llegar a un consenso en la solución del problema [4].

Gamification es el concepto de la aplicación de la mecánica de juego y técnicas de diseño de juegos para involucrar y motivar a las personas a lograr sus metas [5]. A través de Gamification no sólo podemos crear una mentalidad que anima a los estudiantes a probar cosas nuevas, a no tener miedo de fracasar [6], sino que también permite a los estudiantes a participar en experiencias agradables con el propósito del aprendizaje. Una implementación innovadora para el aprendizaje es el uso de juegos serios (Serious Games, SG) comúnmente motivada por la necesidad de educar, formar o informar sobre un tema determinado [7]. La potencia de los juegos serios como una herramienta para el aprendizaje se basa fundamentalmente en la capacidad de equilibrar el entretenimiento, la interactividad y la re-jugabilidad de los juegos típicos con los objetivos de aprendizaje de un objetivo educativo determinado.

Existen diversas revisiones acerca del gamification como Hamari, Koivisto & Sarsa [8] que revisaron estudios empíricos sobre gamification, para esto crearon un marco para examinar los efectos de su aplicación mediante la elaboración de definiciones y las motivaciones que esta provoca obteniendo resultados positivos de la aplicación de gamification. Por otra parte Seaborn & Felsb [9] presentaron un estudio sistemático sobre el uso de gamification en exámenes teóricos y trabajos de investigación relacionados con los sistemas interactivos y los participantes. Diseñaron entendimientos teóricos actuales de gamification y comparaciones con los enfoques relacionados, incluyendo diversos tipos de juegos y el diseño gameful. Los resultados obtenidos muestran que un nivel de conceptualización de gamification creciente en investigaciones con participantes empíricos. Otra revisión en la literatura fue hecha por Fui-Hoon Nah et al. [10] sobre gamification en el contexto educativo y del aprendizaje, en la revisión identifican varios elementos de diseño del juego como puntos,

niveles/etapas y retroalimentación del progreso del juego para identificar resultados obtenidos en el alumno.

El presente trabajo tiene como objetivo hacer una revisión de las técnicas gamification implementadas en el desarrollo de aplicaciones educativas y la implementación o interacción con otras tecnologías como es el caso de Serious games. La importancia de su investigación es debido a que gamification es un término genérico informal para el uso de elementos de juego en sistemas que no son de juego que tienen como objetivo mejorar la experiencia del usuario y la participación de los usuarios [11]. Además, con este trabajo se pretende ofrecer un enfoque innovador para el aprendizaje debido a las nuevas tecnologías y las nuevas aplicaciones que están surgiendo continuamente.

2. Estado del arte

En los últimos años, se han propuesto varios estudios con el objetivo de mejorar la educación bajo el uso de técnicas de inteligencia artificial. La mayor parte de éstos se han centrado en el uso de los juegos serios y aplicaciones gamification en una variedad de contextos. Sin embargo, aún no se reporta el uso de aplicaciones gamification en combinación con otras tecnologías en contextos educativos. En esta sección, se presenta un conjunto de trabajos relacionados con el uso de aplicaciones gamification. Estos trabajos se agruparon de acuerdo con el tipo de tecnología desarrollada: 1) Gamification, 2) Juegos Serios, 3) ITIL y 4) TEL.

2.1. Gamification

Simões, Díaz-Redondo, & Fernández-Vilas [12] encontraron características distintivas de los buenos juegos, con el fin de entender lo que tiene sentido aplicar en los procesos de enseñanza, para esto se utilizó una plataforma online llamada schooools.com, la cual es un ambiente de aprendizaje colaborativo y social. El objetivo fue extraer los mejores elementos del juego para adaptarlos y utilizarlos en el proceso de enseñanza. Por su parte Armstrong et al. [13] definió el concepto de gamification como una forma de aprovechar mecanismos de motivación para aumentar la inversión individual en un sistema, proceso o recurso; es por ello que se aplicaron estos mecanismos para aumentar el compromiso con el catálogo de una editorial, la implementación de gamification satisface los objetivos de negocio y de audiencia. Una forma de aplicarlo es por medio de la motivación extrínseca en donde se participa únicamente a causa de factores externos no relacionados con la naturaleza de la actividad en sí, tales como la oportunidad de ganar un premio o mediante la amenaza de un castigo. Lubin et al. [14] propuso que gamification ofrece una forma didáctica distinta de diseñar una oportunidad para atraer personas y hacer relevante una situación o instrucción a través de la implementación de gamification en entornos de formación educativa. Por su parte Kapp et al. [15] enfatizó la gamification como una forma de ofrecer una mejora del aprendizaje, si gamification se aplica correctamente podría revolucionar el enfoque de una organización a la formación y el desarrollo. Los beneficios podrían incluir más productividad, menos costos de los procedimientos y la satisfacción general con las iniciativas de formación y desarrollo de la institución u

organización. Robson, Plangger, Kietzmann, MacCarthy & Pitt [16] discutieron cómo gamification mejora la forma en que las empresas atraen a los clientes y empleados siguiendo un enfoque Business Horizons basado en los principios de gamification para atraer nuevos clientes, propiciar la participación de los empleados en actividades de gamification todo esto con un control de los casos en el cual también se incluyen las experiencias sin éxito.

2.2. Serious games

Obikwelu & Read [17] determinó el grado en que los juegos serios han adoptado el principio pedagógico del constructivismo en el enfoque del aprendizaje. El objetivo del constructivismo en los juegos serios es que los estudiantes sean capaces de completar el aprendizaje de una manera independiente. Pereira et al. [18] estableció una taxonomía para analizar la situación actual (tendencias y deficiencias), las barreras y facilidades para la adopción de juegos serios en el aprendizaje personal, social y ético; la taxonomía es una herramienta fundamental para crear una estructurada instantánea de un estado actualizado, las tendencias y las deficiencias de un campo de interés. La taxonomía tiene dos temas de clasificación de orientación: el área de la actividad y el objetivo del juego. Carozzinoa, Evangelistaa, Brondia, & Loren [19] presentaron SONNA, un proyecto de investigación con el objetivo de analizar el impacto de las redes sociales, Web 2.0 y multimedia interactivos como herramientas para el aprendizaje. El proyecto de investigación SONNA propone investigar nuevas metodologías de aprendizaje potenciado por la tecnología. La hipótesis que se presenta es que la experiencia dentro de estos contextos mixtos generaría procesos de aprendizaje diferentes de los producidos en contextos tradicionales. Raybourn et al. [20] presentó un nuevo paradigma para la formación y la educación más eficaz y escalable llamado aprendizaje Transmedia. El aprendizaje Transmedia se define como el sistema escalable de mensajes que representa un relato o una experiencia básica que se desarrolla a partir del uso de múltiples medios de comunicación, este paradigma de aprendizaje ofrece un uso más eficaz de los juegos serios para la formación y la educación. En Barbosa & Silva [21] se analizó la importancia de Serious Games y se detallan las fases de desarrollo de un juego serio desarrollado para la Web, utilizando la tecnología WebGL. Con esta tecnología, los desarrolladores crean entornos 3D convincentes y juegos de vídeo en 3D que se pueden acceder por casi todas las personas que tiene una conexión a Internet.

2.3. Technology enhanced learning

La introducción de Tecnologías para la Educación (TEL) promete una mayor personalización y una mayor productividad, Laurillard et al. [22] examinaron diferentes enfoques para comparar el costo de tiempo de enseñanza de TEL contra los métodos tradicionales, concluyendo que el modelo de costos y beneficios dentro de una institución es la forma más exacta de la comprensión de cómo los profesores utilizan la tecnología para alcanzar el nivel de productividad deseado. Redeker [23] conceptualizó una taxonomía didáctica de objetos de aprendizaje y un enfoque didáctico de metadatos para la facilitación de patrones de navegación de instrucción reutilizables. El objetivo del desarrollo fue implementar estructuras genéricas de navegación basadas en el

concepto de jerarquía y tipificación didácticas con el fin de admitir la escritura y los procesos de aprendizaje.

2.4. Information technology infrastructure library

Chung, Rainer & Lewis [24] examinaron el impacto de los cuatro componentes de la flexibilidad de la infraestructura de TI (compatibilidad, conectividad, modularidad, y el personal de TI) en la alineación TI-negocio estratégico y el grado en que varias aplicaciones se implementan dentro de una organización. El estudio refuerza la importancia de ITIL para las organizaciones como una fuente de ventaja competitiva sostenible. Vinogradsky [25] enfatizó que ITIL es ampliamente reconocido como un sistema de mejores prácticas para los departamentos de TI y mesas de servicio es por ellos que está siendo adoptado por las organizaciones de todo el mundo, ya que proporciona un marco basado en la práctica que integra y centraliza la gestión de incidencias, gestión de problemas, gestión de archivos de configuración y gestión del cambio dentro de una plataforma de mesa de servicio. Mohamed, Ribière, O'Sullivan & M. A. Mohamed [26] implementaron ITIL v2.0 en el marco de trabajo de una empresa de gestión de ingeniería a través de principios de gestión de conocimientos incorporados, este marco establece las actividades que son imprescindibles para la mejora de la implementación de ITIL siguiendo con un enfoque de pensamiento para añadir y mantener una ventaja competitiva a través de un enfoque cliente-conciencia y apoyado en el proceso de toma de decisiones a lo largo de la implementación de ITIL, el resultado es un marco de trabajo completo para que las organizaciones alcancen la eficacia y la innovación en toda la implementación.

3. Análisis

Actualmente se considera que las aplicaciones tecnológicas que utilizan juegos pueden explorar un objetivo de aprendizaje o formación profesional. [5] Desde la perspectiva de la innovación, conceptos tales como los juegos serios y gamification son los más interesantes y valiosos en este dominio. Desde un punto de vista académico, los juegos serios se estudian en diferentes disciplinas, como la informática, psicología, ciencias de la educación o la comunicación. A pesar del crecimiento en el uso de juegos serios, muchas preguntas de investigación siguen siendo exploradas y la literatura sigue siendo dispares, tales como las relativas a su diseño, selección, adopción, uso y evaluación. La gamification se considera hoy en día como un motor esencial de la innovación en este campo [5].

Lameras et al. [27] propuso una clasificación de juegos serios con base en las características de diseño de aprendizaje y las propiedades del juego. Este estudio proporciona orientación y apoyo a los profesores de instrucción y los diseñadores de juegos para diseñar, planificar y utilizar los juegos serios para un tema o módulo completo, de esta manera propone el diseño del aprendizaje como modalidad fundamental en la arquitectura de diseño de juego; los atributos del aprendizaje establecidos en la propuesta son aprendizaje colaborativo, individual, transmisión de información y discusión y argumentación; el estudio también propone diversos atributos de los juegos los cuales se toman como base para establecer las siguientes

categorías juegos de reglas, juegos de objetivos y opciones, juegos de tareas y retos, juegos de colaboración y competencia y juegos de regeneración/evaluación.

Las actividades de aprendizaje en juegos tiene el objetivo de obtener resultados de aprendizaje establecidos por el instructor. El resultado de algunas actividades se utiliza como inicio para otras personas como resultado del seguimiento en el juego que puede adaptarse durante la ejecución de la actividad de aprendizaje. Una actividad de aprendizaje basado en el juego a diferencia del contenido del juego es el propósito central para trabajar en el diseño del aprendizaje, el cual que tiene raíces históricas en el campo más amplio de diseño instruccional [28].

Lameras et al. [27] percibe que en el juego las actividades de aprendizaje son una acción situada - que está influenciado por las creencias y valores que poseen los maestros como los diseñadores de juegos en contextos concretos de la práctica, como un proceso iterativo emergente que se produce durante y antes a la orquestación de la actividad de aprendizaje en el juego. Para esto describe una clasificación de las actividades de aprendizaje que se presenta a continuación en la tabla 1:

Tabla 1. Tipos de aprendizaje y sus actividades de aprendizaje.

Tipos de actividades de aprendizaje	Actividades de aprendizaje
<p>Transmisión de información Son actividades que permiten reflejar la información en el aprendizaje del jugador [29][30].</p>	<ul style="list-style-type: none"> • Conferencias, notas de lecturas, diapositivas. • Memorización de conceptos. • Esquemas • Ejemplos • Declaraciones incompletas • Resumen • Audios
<p>Individual Son actividades para desarrollar en solitario y adquirir conocimiento [27].</p>	<ul style="list-style-type: none"> • Web-quest (Búsqueda y recuperación de información) • Resolución de ejercicios • Experimentación científica • Reflexiones • Simulaciones • Modelo • Juegos de rol • Investigación • Determinación de pruebas • Análisis de la evidencia • Formulación de pruebas • Conectar explicaciones sobre conocimientos
<p>Colaborativo Actividades que permiten adquirir conocimiento involucrando a varios</p>	<ul style="list-style-type: none"> • Reuniones • Proyectos en grupo • Web-quest en grupo • Rangos y reportes

Tipos de actividades de aprendizaje	Actividades de aprendizaje
personas para obtener un resultado en común [27].	<ul style="list-style-type: none"> • Grupo de estudiantes que se presentan preguntas entre ellos • Simulaciones en grupo • Resolución de problemas en grupo • Recopilación de datos en grupo • Análisis de datos en grupo • Reflexiones en grupo
<p>Discusión y argumentación Actividades que permiten la retroalimentación [27].</p>	<ul style="list-style-type: none"> • Discusiones guiadas (tema de discusión proporcionado por el maestro) • Discusiones abiertas (tema de discusión proporcionado por los estudiantes) • Opciones: datos sobre los eventos y varias opciones para hacer comentarios por parte de los estudiantes. • Debates.

Beetham [29] definió la actividad de aprendizaje como una "interacción específica de estudiantes con otras personas que utilizan herramientas y recursos específicos, orientados a resultados específicos". Es por ello la importancia del diseño de las actividades de aprendizaje en el desarrollo de juegos serios y aplicaciones gamification para cumplir con el objetivo por el cual fue desarrollado el juego.

Otro aspecto importante a considerar en los juegos son sus atributos es por ello que son objeto de estudio de varios trabajos como Lundgren & Bjork [31] que definieron los atributos del juego como una forma de resumir las reglas del juego, aunque deja en duda si solo abarca las reglas que definen la mecánica o abarca características utilizadas en el diseño [27]. Desde una perspectiva educativa Cook [32] interpretó los atributos de juego dando énfasis a las propiedades de retroalimentación al tiempo que reconoce las relaciones entre las normas y los atributos del jugador. Los atributos del juego deben ser considerados en el diseño del juego para garantizar un equilibrio entre los desafíos y las habilidades necesarias para alcanzar los objetivos.

Con base en los atributos de un juego, Lamerás [27] los clasificó en categorías de juegos dependiendo de sus atributos relevantes. Se realizó un intento para asignar categorías de juego generales comprendiendo los atributos en el juego que se utilizan para la creación de instancias de atributos de juego en la práctica educativa; por ejemplo, las reglas se realizan a través de puntuación. El autor hace esta clasificación debido a que no hay una taxonomía que clasifique los atributos del juego en categorías específicas, la clasificación la elaboró con el objetivo de ayudar a los diseñadores de juegos y de instrucción. Las categorías que presenta Lamerás [27] se describen la tabla 2, las cuales se basan de acuerdo a la identificación de los atributos de juegos en cada una de ellas:

Tabla 2. Categorías de juegos y los atributos de cada categoría.

Categorías de juegos	Atributos de juegos
Juegos de reglas	<ul style="list-style-type: none"> • Niveles de puntuación

Categorías de juegos	Atributos de juegos
	<ul style="list-style-type: none"> • Niveles de movimiento • Temporizadores • Barras de progreso • Instrucciones del juego incluyendo las condiciones de la victoria
Juegos de objetivos y opciones	<ul style="list-style-type: none"> • Diario de juego • Misiones • Tarjetas objetivas • Historias • Diálogos anidados • Rompecabezas • Avatares / NPCs
Juegos de tareas y desafíos	<ul style="list-style-type: none"> • NPC descripción de las tareas basadas • Barras de progreso • Múltiples opciones para elegir • Tareas principales • Tareas de rama • Rompecabezas • Puntos de investigación • Requisitos
Juegos de colaboración y competencia	<ul style="list-style-type: none"> • Juego de roles • Colaboración de la comunidad • Sentido épico • Bonos • Temporizadores • Descubrimiento líder • Concursos • Puntuación • Inventarios • Tablas
Juegos de retroalimentación y evaluación	<ul style="list-style-type: none"> • Consejos de juego • Personajes • Niveles de juego • Obtención y pérdida de vidas • Barras de progreso • Cuadros de mando

A continuación se presenta una tabla comparativa de los trabajos analizados en este estudio los cuales pertenecen a diversas tecnologías como es el caso de gamification, juegos serios, TEL e ITIL con el objetivo de encontrar qué tipo de actividades de aprendizaje se encuentra en cada uno y qué atributos de juego fueron encontrados en

cada trabajo y con base en atributo del juego encontrado a qué clasificación del juego pertenece ese trabajo. Cabe mencionar que debido a que son varias tecnologías analizadas no todos los trabajos cumplen los aspectos del análisis.

Tabla 3. Análisis de identificación de la categoría y atributos del juego de los trabajos seleccionados para este estudio.

Artículo	Objetivo	Tipos de actividades de aprendizaje	Categoría / Atributo del juego
Simões, Díaz-Redondo, & Fernández-Vilas [12]	Implementación de una plataforma online.	Transmisión de información	Objetivos y opciones -Descripción de la tarea. -Descripción del contenido. -Repetición del objetivo.
Armstrong et al. [13]	Estudio de la motivación extrínseca.	Individual	Objetivos y opciones -Misiones. -Jornadas de juego. -Niveles de juego.
Lubin et al. [14]	Implementación en entornos educativos.	Colaborativo	Colaboración y competencia -Colaboración de comunidad.
Kapp et al [15]	Implementación en la formación y desarrollo de organizaciones.	Transmisión de información	Tareas y desafíos -Descripción de la tarea. -Descripción del contenido.
Robson, Plangger, Kietzmann, MacCarthy & Pitt [16]	Implementación para atraer nuevos clientes y empleados.	Colaborativo	Colaboración y competencia -Colaboración de comunidad. -Descubrimiento -Niveles de juego -Roles
Obikwelu & Read [17]	Determinación del grado de adoptado en el principio del constructivismo en un enfoque educativo.	Colaborativo	Colaboración y competencia -Colaboración de comunidad.

Artículo	Objetivo	Tipos de actividades de aprendizaje	Categoría / Atributo del juego
Pereira et al. [18]	Desarrollo de una taxonomía de objetivos de aprendizaje.	Transmisión de información	Tareas y desafíos -Descripción de la tarea.
Carrozzinoa, Evangelistaa, Brondia, & Loren [19]	Presenta SONNA, plataforma para investigar nuevas métricas de aprendizaje.	Colaborativo	Colaboración y competencia -Colaboración de comunidad. -Tablas de clasificación.
Raybourn et al. [20]	Presenta un nuevo paradigma de aprendizaje llamado Transmedia.	Colaborativo	Colaboración y competencia -Colaboración de comunidad. -Tablas de clasificación.
Barbosa & Silva [21]	Detalla las fases de desarrollo de un juego serio desarrollado para la web	Transmisión de información	Tareas y desafíos - Descripción de la tarea. - Descripción del objetivo.
Laurillard et al. [22]	Modelo orientado a costos en la enseñanza	Transmisión de información	Tareas y desafíos - Descripción de la tarea.
Redeker [23]	Implementación de patrones de navegación	Individual	Tareas y desafíos - Puntos de investigación. - Requisitos del estudio. - Niveles de juego.
Chung, Rainer & Lewis [24]	Implementación ITIL en organizaciones	Colaborativo	Colaboración y competencia - Colaboración. - Clasificación.

4. Discusión

La tabla 3 presenta los trabajos analizados en este estudio sometidos a la clasificación de Lameris [27] tomando en cuenta los tipos de actividades de

aprendizaje, los atributos de los juegos y la clasificación establecida, los trabajos presentados están elaborados en diversas tecnologías como gamification, juegos serios, Information technology Infrastructure Library y Technology enhanced learning con el objetivo encontrar atributos y tipos de actividades para encontrar un punto de referencia para su posterior integración a gamification.

El objetivo de las actividades de aprendizaje en juegos es obtener un resultado establecido es por ello que se sometió a una evaluación los trabajos previamente seleccionados tomando como base la clasificación de Lameris [27] para identificar dependiendo de las actividades que se desempeñaron en cada uno de ellos; para hacer la clasificación de a que tipo de actividades pertenece cada trabajo y de igual manera analizar si cumplían con el propósito para el que fueron diseñados.

Después de identificar los tipos de actividades de aprendizaje se identificaron los atributos del juego que estaban presentes en cada trabajo, cabe mencionar que aunque no todos los trabajos tienen la aplicación de un juego o alguna técnica relacionada se sometieron al análisis para identificar aquellos factores que estén presentes en su desarrollo para posteriormente como trabajo a futuro integrar las diferentes tecnologías implementadas en aplicaciones gamification con el objetivo de fortalecer su aplicación y abarcar otras tecnologías emergentes.

El resultado de los tipos de actividades de aprendizaje presentes en los trabajos analizados se muestra en la Figura 1.

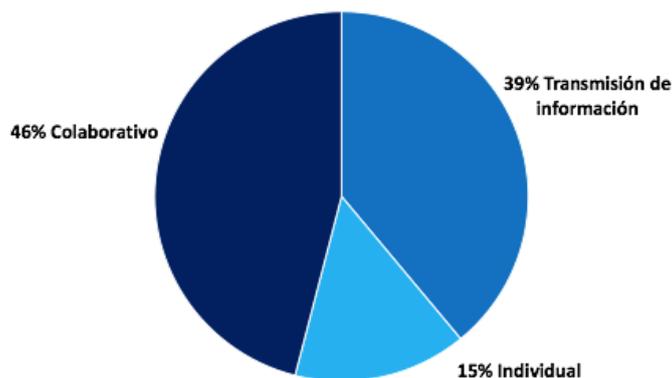


Fig. 1. Tipos de actividades de aprendizaje encontradas en los artículos seleccionados.

El resultado obtenido en trabajos de gamification presentan la transmisión de información, individual y colaborativo dependiendo del contexto de cada trabajo, al igual que los trabajos relacionados con los juegos serios que están presentes estos tipos de actividades de aprendizaje transmisión de información, individual y colaborativo; mientras tanto en las otras tecnologías analizadas como TEL presentó la transmisión de información y actividades del tipo individual e ITIL presentó el tipo de actividad colaborativo. Las aplicaciones gamification y los juegos serios al estar ligados con el uso e implementación de juegos se hizo una identificación de todos los tipos de actividades de aprendizaje ya que pertenecen a la misma área, mientras que TEL e ITIL son tecnologías que aunque no implementan técnicas de juego se identificaron actividades de aprendizaje y atributos del juego para en un futuro por medio de este punto en común se puedan implementar en aplicaciones gamification.

Los atributos del juego presentes en los trabajos de aplicaciones gamification son la descripción de tareas, descripción de contenido, repetición del objetivo, misiones, jornadas de juego, niveles de juego y la colaboración de comunidad; las aplicaciones gamification emplean técnicas del juego es por ello que se pueden identificar mas de un atributo en cada una de las aplicaciones. En los juegos serios los atributos presentes son la colaboración de la comunidad, tablas de clasificación, descripción de la tarea y descripción del objetivo; los juegos serios son juegos cuyo objetivo esta definido desde su diseño es por ello que la descripción de la tarea y del objetivo están definidas desde su diseño. Por otra parte los trabajos relacionados con TEL presentaron los atributos de juego descripción de la tarea, niveles, requisitos de estudio y puntos de investigación. ITIL presentó la colaboración y clasificación.

Una vez identificados los atributos de los juegos, se realizó una clasificación de acuerdo a los atributos a qué categoría de juego pertenecen los trabajos, estos resultados se observan en la Figura 2.

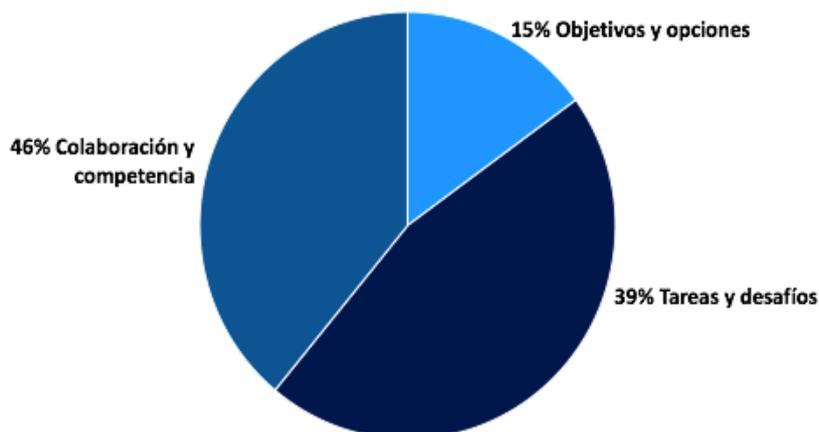


Fig. 2. Categorías de juegos encontradas en los artículos seleccionados.

Como resultado de la revision sistematica se obtuvo que la categoria de juegos que mas se implementa en aplicaciones gamification y juegos serios es la competencia y colaboracion, la cual se implementa en casi la mitad de los trabajos analizados, seguido de la categoria de Tareas y desafios y la categoria de Objetivos y opciones. Cabe destacar que en ninguno de los trabajos se encontraron atributos de juego de la categoria de Discusion y argumentación.

5. Conclusión y trabajo a futuro

La implementación de Gamification y juegos serios resulta ser innovadora para el aprendizaje, Gamification es el concepto de la aplicación de la mecánica de juego y técnicas de diseño de juegos para involucrar y motivar a las personas a lograr sus metas proporcionando experiencias agradables con el objetivo del aprendizaje. Mientras que los juegos serios son una herramienta potente para el aprendizaje basados fundamentalmente en la capacidad de equilibrar el entretenimiento, la interactividad y

la re-jugabilidad con los objetivos de aprendizaje de un objetivo educativo determinado. El presente trabajo presenta una revisión sistemática de trabajos actuales en el área de gamification y juegos serios sometiéndolos a una identificación de los tipos de actividades de aprendizaje presentes en casa trabajo así como que atributos de juego presenta y en base a ellos a que clasificación de juegos pertenece.

Como trabajo a futuro se considera realizar una investigación de más trabajos relacionados con las tecnologías de este estudio con el fin de analizar de manera detallada que otros tipos de actividades de aprendizaje están presentes así como que atributos de juego abarcan. De igual manera se pretende abordar otras clasificaciones de juegos propuestas como la clasificación de Caillois [33], Lavega [34] entre otros. Así como investigar y analizar otros atributos de juego como los propuestos por Caillois [33].

Agradecimientos. Este trabajo es apoyado por el Consejo Nacional de Ciencia y Tecnología (CONACYT), Tecnológico Nacional de México (TecNM) y la Secretaría de Educación Pública (SEP) a través de PRODEP.

Referencias

1. Prensky, M.: *Digital Game-Based Learning*. Minnesota: Paragon House Ed edition (2005)
2. Ching-Hsue, C., Chung-Ho, S.: A Game-based learning system for improving student's learning effectiveness in system analysis course. *Procedia - Social and Behavioral Sciences* Vol. 31, pp. 669–675 (2012)
3. Pivec, M.: Game-based learning in universities and lifelong learning: "UniGame: Social Skills and Knowledge Training" game concept. *Special Issue of J.UCS Human Issues in Implementing e-learning technology*, Vol. 10, No. 1, pp. 14–26 (2004)
4. Bouras, V., Igglesis, V., Kapoulas, I., Misedakis, O., Dziabenko, A., Koubek, M.: Game-based learning using web technologies. *Journal of Intelligent games and Simulation*, Vol. 3, No. 2, pp. 67–84 (2004)
5. Burke, B.: Gartner Redefines Gamification. 4 de Abril de 2014. http://blogs.gartner.com/brian_burke/2014/04/04/gartner-redefines-gamification/ (último acceso: 31 de Marzo de 2016).
6. Chung-Ho, S., Ching-Hsue, C.: A Mobile Game-based Insect Learning System for improving the learning achievements. *Procedia - Social and Behavioral Sciences*, No. 103 pp. 42–50 (2013)
7. Michael, D., Chen, S.: *Serious Games: Games that educate, train and inform*. Mason: Cengage Learning PTR (2005)
8. Hamari, J., Koivisto, J., Sarsa H.: Does Gamification Work? — A Literature Review of Empirical Studies on Gamification. *Proceeding at 47th Hawaii International Conference on System Science*, pp. 3025–3034 (2014)
9. Seaborn, K., Felsb, D.I.: Gamification in theory and action: A survey. *International Journal of Human-Computer Studies*, Vol. 74, pp. 14–31 (2015)
10. Fui-Hoon Nah, F., Zeng, Q., Rajasekhar-Telaprolu, V., Padmanabhuni-Ayyappa, A., Eschenbrenner, B.: Gamification of Education: A Review of Literature. *HCI in Business* Vol. 8527, pp. 401–409 (2014)
11. Pereira, P., Duarte, E., Rebelo, F., Noriega, P.: A Review of Gamification for Health-Related Contexts. *Design, User Experience, and Usability, User Experience Design for Diverse Interaction Platforms and Environments*, pp. 742–753 (2014)

12. Simões, J., Díaz-Redondo, R., Fernandez-Villas, A.: A social gamification framework for a K-6 learning platform. *Computers in Human Behavior*, Vol. 29, pp. 245–253 (2013)
13. Armstrong, D.: The new engagement game: the role of gamification in scholarly publishing. *Learned Publishing*, Vol. 26, No. 4, pp. 253–256 (2013)
14. Lubin, L.: The Gamification of Learning and Instruction Field Book. *New Horizons in Adult Education & Human Resource Development*, Vol. 28, No. 1, pp. 58–60
15. Kapp, K.: *Gadgets, games, and gizmos for learning* (1st Ed.). San Francisco, CA: John Wiley (2007)
16. Robson, K., Plangger, K., Plangger, J.H., McCarthy, I., Pitt, L.: Game on: Engaging customers and employees through gamification. *Business Horizons* (Elsevier) Vol. 59, pp. 29–36 (2016)
17. Obikwelu, C., Read, J.: The serious game constructivist Framework for children's learning. *Procedia Computer Science*, Vol. 15, pp. 32–37 (2012)
18. Pereira, G., Brisson, A., Prada, R., Paiva, A., Bellotti, F., Kravcik, M.: Serious Games for Personal and Social Learning & Ethics: Status and Trends. *Procedia Computer Science*, Vol. 15, pp. 53–65 (2012)
19. Carrozzino, M., Evangelista, C., Brondia, R., Loren, C.: Social Networks and Web-based Serious Games as Novel Educational Tools. *Procedia Computer Science*, pp. 303–306 (2012)
20. Raybourn, E.M.: A new paradigm for serious games: Transmedia learning for more effective training and education. *Journal of Computational Science*, pp. 471–481 (2014)
21. Barbosa, A.F.S., Silva, F.G.M.: Serious Games - Design and Development of OxyBlood. *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology, ACE*, (2010)
22. Laurillard, D.: Modelling benefits-oriented costs for technology enhanced learning. *Higher Education* (Springer) Vol. 54, No. 1, pp. 21–39 (2007)
23. Redeker, G.H.J.: An Educational Taxonomy for Learning Objects. *Proceedings of the The 3rd IEEE International Conference on Advanced Learning Technologies (ICALT'03)*, (2003)
24. Chung, S.H., Rainer Jr., R.K., Lewis, B.R.: e Impact of Information Technology Infrastructure Flexibility on Strategic Alignment and Application Implementations. *Communications of the Association for Information Systems*, Vol. 11, pp. 191–206 (2003)
25. Vinogradsky, V.: The ITIL Framework: And How It Can Improve Your Service Desk. *Alloy software*, pp. 16–18 (2008)
26. Mohamed, M.S., Ribière, V.R., O'Sullivan, K.J., Mohamed, M.A.: The re-structuring of the information technology infrastructure library (ITIL) implementation using knowledge management framework. *VINE Journal of Information and Knowledge Management Systems (VINE)*, Vol. 38, No. 3, pp. 315–333 (2008)
27. Lameris, P.: Essential Features of Serious Games Design in Higher Education. *Society for research into higher education*, pp. 3–22 (2015)
28. McLean, P., Scott, B.: Competencies for Learning Design: A Review of the Literature and a Proposed Framework *British Journal of Educational Technology*, Vol. 42, No. 4, pp. 557–572 (2011)
29. Beetham, H.: Review: Design for learning programme phase 2. . Review of learning design as part of the JISC's Design for Learning programme. Available at: http://www.jisc.ac.uk/whatwedo/programmes/elearningpedagogy/designlearn.a_spx. [Accessed 28 April 2011] (2008)
30. Gunter, G., Kenny, R., Vick, E.: A Case for a Formal Design Paradigm for Serious Games. *The Journal of the International Digital Media and Arts Association*, Vol. 3, No. 1, pp. 1–19, doi: citeulike-article-id:9301918 (2006)
31. Lundgren, S., Bjork, S.: Game mechanics: Describing computer-augmented Games in Terms of Interaction. Paper presented at the TIDSE, Available online at:

- http://www.itu.dk/stud/speciale/worlddomination/files/rikke/rh/speciale/staffan_docs/mechanics.pdf [last accessed 23 February 2015] (2003)
32. Cook, D.: What are game mechanics Available at: <http://www.lostgarden.com/2006/10/what-are-game-mechanics.html> [Last accessed 23 February 2015]: Lostgarden (2006)
 33. Caillois, R.: Teoría de los Juegos. 1ª. Barcelona: Seix Barral (1958)
 34. Lavega, P.: La clasificación de los juegos. http://www.praxiologiamotriz.inefc.es/PDF/clasificacion_juegos__deportes.pdf (2013)

Identificación automática de marcadores argumentativos en discursos políticos

Felipe Ojeda-Cruz¹, Noé Alejandro Castro-Sánchez¹,
Héctor Jiménez-Salazar²

¹ Departamento de Ciencias de la Computación CENIDET,
México

² Departamento de Tecnologías de la Información, Unidad Cuajimalpa UAM,
México

{felipe.ojeda,ncastro}@cenidet.edu.mx, hgimenezs@gmail.com

Resumen. En este artículo se presenta el desarrollo de un método para identificar marcadores argumentativos, al que denominamos “inducción de patrones”. Método que a partir de uno o varios patrones, encuentra otros. Los patrones obtenidos se aplicaron a un conjunto de documentos sobre política para identificar marcadores de argumentos. Los resultados obtenidos arrojan una precisión del 92% y una cobertura del 25% en la extracción de patrones contextuales de los marcadores.

Palabras clave: marcadores argumentativos, Capaldi, Freeling, patrones sintácticos, POS, ete-toolkit, morfosintáctico, identificación de argumentos.

Automatic Identification of Argumentative Markers in Political Speeches

Abstract. This article describes the development of a method to identify argumentative markers, which we call “induction patterns”. Method from one or more patterns, find others. The patterns obtained are applied to a set of policy documents for to identify markers of arguments. The end result is used to identify arguments with a regular expression. The results show a precision of 92% and 25% coverage in the extraction of contextual patterns of markers.

Keywords: argumentative markers, Capaldi, Freeling, syntactic patterns, POS, ete-toolkit, morphosyntactic, identification of arguments.

1. Introducción

En el marco de la lógica, los argumentos “son conjuntos de razones o de pruebas para apoyar una conclusión”[1], de igual manera los argumentos intentan persuadir y

convencer a terceras personas de lo que se está proponiendo, es por eso que los argumentos son una parte importante en nuestro lenguaje diario. Argumentamos por muchas razones, por ejemplo, para convencer a nuestro interlocutor sobre algún documental o sobre la importancia de comer saludable, entre muchas otras cosas más.

La naturaleza de los argumentos es muy compleja, hay argumentos implícitos y explícitos, estos últimos hacen uso de palabras claves (marcadores) que facilitan la identificación de los argumentos. Los marcadores consisten en palabras claves que funcionan como una especie de conjunción o de introductores entre proposiciones.

La identificación de argumentos en el área de Inteligencia Artificial es un tema relativamente nuevo, los primeros avances computacionales fueron en el 1st Argument Mining workshop at ACL 2014 y 2st Argument Mining workshop at ACL 2015, en ellos se utilizan diferentes técnicas de procesamiento de lenguaje natural como los n-gramas.

Por otro lado, en el presente trabajo se desarrolló un método de inducción de patrones para identificar más marcadores argumentativos y así, permitir identificar más argumentos explícitos; utilizando el analizador sintáctico Freeling[2], ya que es un analizador con una gran precisión morfosintáctica.

Estudiar los argumentos políticos es una tarea interesante porque el problema con los argumentos de cualquier proyecto público es saber fijar qué hay de fundamento y qué hay de excusa. El argumento que se proporciona al público es la reflexión de la toma de decisiones del gobierno y en que medida lo contado es solo la reelaboración de esas ideas, que pueden servir de justificación, para hacer más atractivo o al menos más tolerante a los ciudadanos[3]. Los discursos políticos llevan una estructura de redacción, porque llevan argumentos lógicos, claros y bien definidos; y así son más persuasivos[4], así como también guiar a los ciudadanos a reflexionar como los políticos desean.

2. Idetificación de argumentos con marcadores

En este artículo nos centramos únicamente en la identificación de argumentos que contengan marcadores de razón y de conclusión, es por ello que se tomó la tarea de investigar cuáles son los marcadores y cómo identificarlos. En [5] se menciona el uso de palabras claves de premisas y conclusión para identificar el argumento en un discurso general. De igual forma plasma algunas ideas y reglas que facilitan esta identificación. Algunas palabras que introducen conclusión son: “como resultado”, “de ahí que”, “en consecuencia”, “se desprende que”, “llegamos a la conclusión”, “por consiguiente” y sinónimos. Algunas palabras que introducen razones son: “puesto que”, “ya que”, “como”, “en tanto que”, “dado que”, “por cuanto”, “viendo que” y sinónimos.

Ya teniendo un conjunto de marcadores se analizaron algunos párrafos que son argumentos. Cabe destacar que el primer paso para identificar argumentos es identificar la proposición que funciona como conclusión[1], y si se encuentra, es que el párrafo es un argumento y se compone de dos partes (premisas y conclusión).

Sin embargo, un párrafo puede que tenga más de una conclusión, y que dicha conclusión funcione como premisa, es decir, toda proposición puede ser tanto una premisa como una conclusión; a esta serie de argumentos interrelacionados se le conoce como sorites[5].

Para resolver el problema de identificar automáticamente argumentos por marcadores, así como también los sorites; se tomó la decisión de implementar expresiones regulares (regex) para analizar todo el discurso.

La expresión regular que se desarrolló (Fig. 1), consiste en dos grandes partes, que se explicarán a detalle en el siguiente apartado. La expresión regular puede identificar las premisas y conclusiones sin importar su posición en el párrafo, de igual forma no se identifican por su contenido. Esto es importante ya que, por la naturaleza de los argumentos, la conclusión se puede dar en cualquier parte del párrafo.

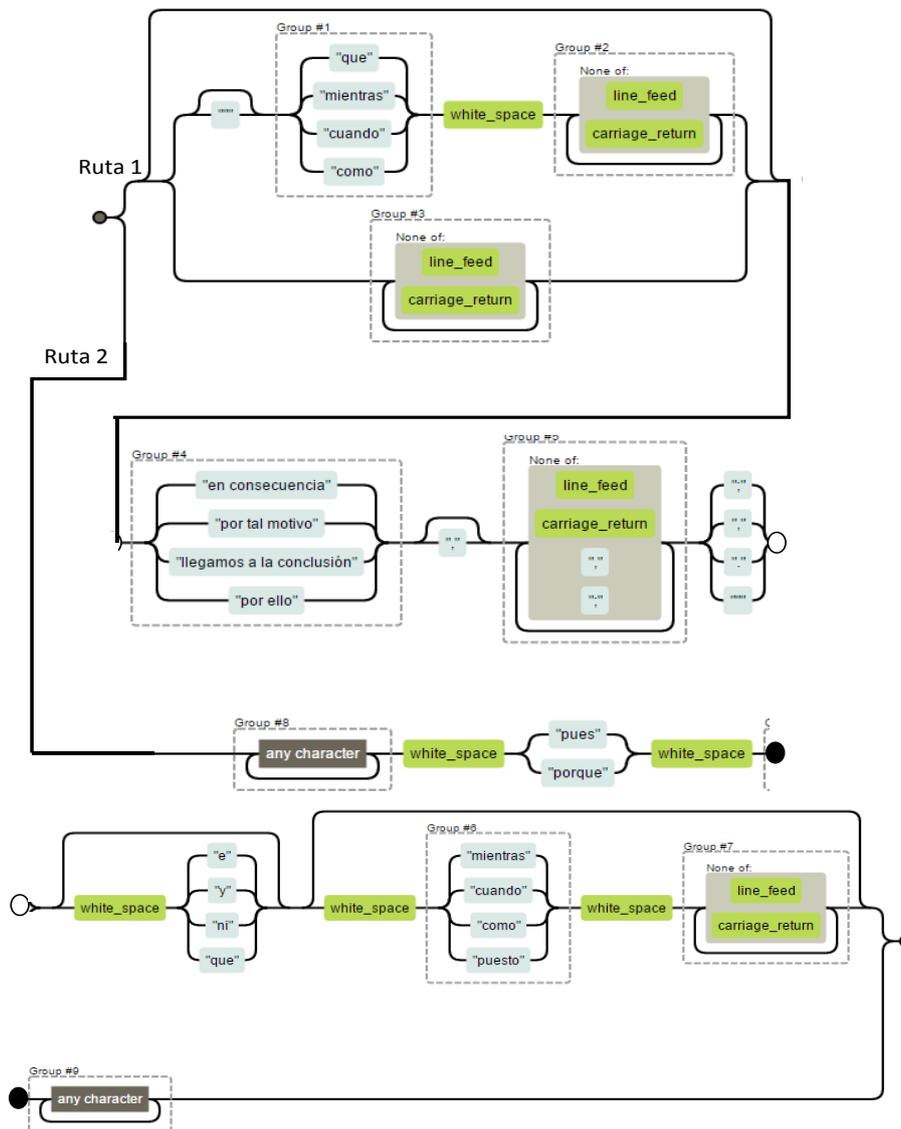


Fig. 1. Diagrama de ferrocarril del regex para identificar argumentos[6].

2.1. Explicación del diagrama

La expresión regular empieza con dos rutas, la ruta 1 es la primera parte y se divide en tres sub-partes más. La primera sub-parte de la expresión regular (grupo del 1 al 3) identifica el conjunto de premisas pero esta identificación es opcional. La segunda parte (grupo del 4 al 5) es la conclusión y no es opcional, de esta forma se pretende encontrar un argumento sobre cualquier de las siguientes estructuras, premisa-conclusión, conclusión-premisa o premisa-conclusión-premisa. Por último tenemos la tercera sub-parte que consiste en identificar premisas, de igual manera opcional con la diferencia que implica, por ejemplo, no iniciar con algunos caracteres.

Aparte de los marcadores de conclusión, existen otras palabras que permiten identificar premisas y conclusiones al mismo tiempo; por mencionar dos casos, “pues” y “porque”. Lo que va después de estas palabras, indica la existencia de una o más premisas, y lo que va antes indica la existencia de conclusión. Por tal motivo, como se aprecia en la figura 1, existen al inicio dos caminos por recorrer; iniciando en el grupo #1 o iniciando desde el grupo #8 (ruta 2).

La expresión regular se divide en nueve grupos, cada uno captura diferentes partes del argumento:

- Grupo #1: Captura el introductor de premisa.
- Grupo #2: Captura la proposición-premisa.
- Grupo #3: Captura la proposición-premisa pero que no cuenta con un introductor de premisa.
- Grupo #4: Captura el introductor de conclusión.
- Grupo #5: Captura la proposición-conclusión.
- Grupo #6: Captura el introductor de premisa que se encuentra después de una conclusión.
- Grupo #7: Captura las proposiciones-premisa que se encuentran después de un introductor de premisa.
- Grupo #8: Captura la proposición-conclusión determinado por los marcadores “pues” o “porque”.
- Grupo #9: Captura la proposición-premisas determinado por los marcadores “pues” o “porque”.

No todos los grupos capturarán al mismo tiempo, o se activan los grupos #1-7 o los grupos #8-9, en los casos del grupo #1-7 se activarán 3 grupos y en los grupos #8-9, siempre se activarán los dos.

Una vez que se identifique el cuerpo del argumento, se aplica de nuevo la expresión regular para identificar si existen otros argumentos dentro del argumento principal y así sucesivamente, hasta que el algoritmo no encuentra ningún introductor de conclusión.

Por otra parte, la consideración de signos de puntuación y/o conjunciones en los argumentos proporciona ciertas pautas que indican proposiciones. Capaldi[5] menciona la importancia de su uso en argumentos por ser utilizados en la mayoría de los casos, por esa razón se consideraron los signos de puntuación en la expresión regular (véase entre los grupos #5 y #6).

2.2. Identificar argumentos interrelacionados

En la mayoría de los casos un argumento está dividido en premisas y conclusión, sin embargo la conclusión puede funcionar como premisa para otro argumento, a esto se le conoce como argumentos interrelacionados o sorites. En teoría, para identificar todos los argumentos que puede haber en un fragmento textual como un párrafo, se aplicaría varias veces la expresión regular, para encontrar los demás argumentos. La razón de aplicarlo de nuevo es que en su defecto encuentran la primera coincidencia y continúan hasta donde finalizó la coincidencia, por lo que no hacen búsqueda superpuesta. Las expresiones regulares tienen la opción de realizar búsquedas superpuestas, sin embargo aplicar esta opción puede desencadenar retroceso catastrófico, es un concepto conocido cuando su motor regresa a un estado anterior para analizar los otros posibles casos en la búsqueda del texto[7].

Por ejemplo, dado el siguiente argumento extraído del sitio www.gob.mx:

“El Programa Nacional de Sustitución de Lámparas Incandescentes por Ahorradoras de Energía busca proteger la economía de las familias más necesitadas; así los hogares reducirán su consumo, y por lo tanto, pagarán menos.”

Al utilizar el método de la expresión regular, identifica dos partes del texto. Lo que está antes de “por lo tanto” son las premisas y lo que está después es la conclusión. Después se tiene como entrada el fragmento de las premisas, la expresión regular identificará de igual forma dos partes; lo que está antes del marcador “así” es la premisa y lo que está después es la conclusión hasta llegar a consumo. De esta forma, se puede identificar todos los argumentos que puede haber en un párrafo.

2.3. Incremento de marcadores

El alcance de la identificación de argumentos con una expresión regular está limitada por el número de marcadores. Así que el siguiente paso que se hizo, fue buscar más sinónimos de los marcadores llegando a una lista más extensa. La tabla 1 indica con asteriscos algunos marcadores que menciona Capaldi, aquellos que no lo tienen son algunos de los encontrados por los recursos *sensagent*, *interglot*, *thefreedictionary* y *RAE*; como también de los autores *Calsamiglia* y *Tuson*[8].

Tabla 1. Muestra de marcadores argumentativos encontrados por algunos recursos.

Marcadores de conclusión		Marcadores de premisas	
<i>por consiguiente</i>	*	<i>al igual que</i>	
<i>se desprende que</i>	*	<i>como</i>	
<i>como resultado</i>	*	<i>igual que</i>	
<i>luego</i>		<i>tan</i>	
<i>llegamos a la conclusión</i>	*	<i>en tanto que</i>	*
<i>entonces</i>		<i>en el supuesto de que</i>	
<i>así pues</i>		<i>en la medida en que</i>	

El número de marcadores de premisas adicionales son 28 y el de conclusiones son 30. Sin embargo es posible que dependiendo del dominio del tema o el tipo de discurso (oral o escrito) se utilicen otros marcadores argumentativos. Ante este hecho, se desarrolló el método para extraer más marcadores, como se verá en el siguiente capítulo.

3. Método de inducción de patrones

El método de inducción se basa en la siguiente consideración. Sea T un texto con etiquetas POS (part of speech) que contiene un marcador de Capaldi, M. Considérese el contexto izquierdo C_i y derecho C_d de M en T; es decir, la cadena C_iMC_d ocurre en T. Si para otro término M' se cumple que C_iM'C_d ocurre en T, diremos que M' satisface el contexto de M y, por lo tanto, puede ser un marcador equivalente a M. Este método tomó como base las ideas presentadas por Marti Hearst en la identificación de hipónimos [9]. Además puede aplicarse este enfoque de inducción en la identificación de definiciones[10].

El método desarrollado consta de dos fases, el método para la extracción automática de patrones contextuales mediante el análisis morfosintáctico, y el método para la búsqueda extendida de expresiones regulares con patrones morfosintácticos.

El método de extracción de patrones requiere como parámetros un conjunto de textos y, y el marcador o marcadores que se quieren analizar. Después del análisis automático se genera una tabla con estadísticas que muestra un conjunto de etiquetas chunking y Eagle, cada tipo de etiqueta se acompaña de la cantidad de veces que apareció en los contextos de la palabra analizada y también se acompaña del porcentaje con respecto al total de textos analizados.

El método siguiente, requiere como entrada una expresión regular, que a su vez implica contener las palabras claves y la sintaxis de estos. Asimismo la expresión regular puede estar conformada de etiquetas Eagle o Chunk, que se identifica del texto o caracteres especiales, porque cada etiqueta tiene al final guion bajo “_”. El resultado es una expresión regular con las etiquetas sustituidas por texto.

3.1. Método de extracción automática de patrones contextuales mediante el análisis morfosintáctico

Freeling posee varios análisis; análisis morfológico, etiquetado POS, análisis de dependencia, análisis sintáctico de poca profundidad, y análisis sintáctico completo. Este último fue el utilizado para identificar los patrones que rodean a los marcadores.

El análisis sintáctico además de contener análisis morfológico contiene las relaciones entre palabras y oraciones, lo que incrementa la posibilidad de extraer patrones más precisos, hasta llegar a un grado de patrones a nivel de sintagmas u oraciones. Estas relaciones entre palabras, Freeling lo llama Chunking Tags (Fragmentos de etiquetas). Por mencionar algunas de las decenas que existen, son: a-fp (Adjetivo, Adjetivo femenino plural), coord. (Conjunción, Conjunción de coordinación), grup-sp (Preposición, fragmento preposición), entre otras[11].

En un análisis sintáctico completo que realiza Freeling, se puede observar en la Fig. 2.

Por ejemplo, se tomó un fragmento textual del discurso “Estado mayor presidencial” del sitio www.gob.mx en el que se menciona lo siguiente[12]:

“El Estado Mayor Presidencial (EMP) es un órgano técnico militar que tiene como misión fundamental proteger al Presidente de la República. Esta responsabilidad es de la más alta importancia para el País, ‘pues’ implica el resguardo de la representación del Estado Mexicano y de la República, ‘por consiguiente’ de la estabilidad, de la gobernabilidad y de la seguridad nacional”.

En el párrafo citado se observa dos marcadores “pues” y “por consiguiente”. Obsérvese en la figura 2 el nodo raíz con la etiqueta “RG” que representa el introductor de conclusión “por consiguiente”, el contexto inmediato a nivel POS (Etiquetas Eagle), son las etiquetas izquierdo “Fc” y derecho “SPS00” y a nivel sintáctico (Etiquetas Chunk), el contexto izquierdo son “Fc” y derecho “sp-de”. Nótese que la etiqueta “sp-de” representa el fragmento textual “de la estabilidad”.

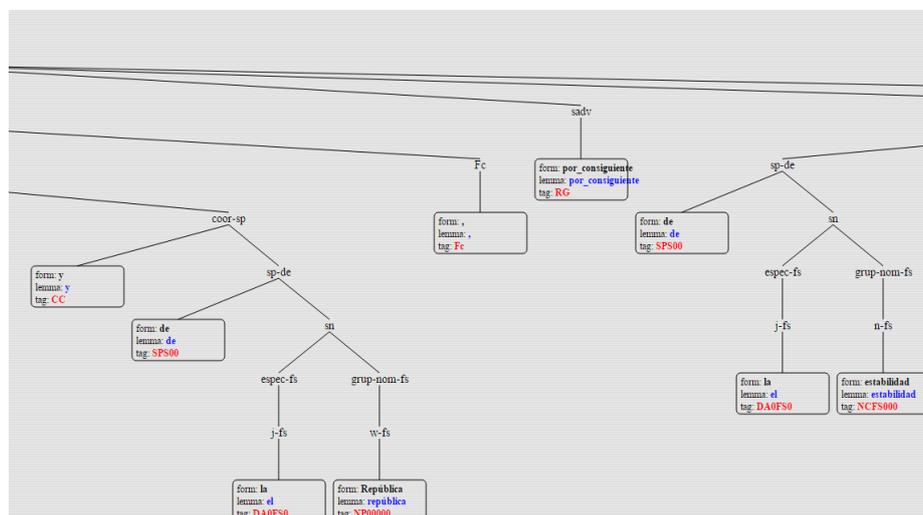


Fig. 2. Representación parcial de un árbol sintáctico del párrafo citado.

Sin embargo, en la mayoría de las situaciones el contexto izquierdo de una palabra puede tener más de una etiqueta Chunk como sucede con la “,”. Analizando el lado izquierdo tenemos que “w-fs, grup-nom-fs, sn, sp-de, coor-sp” son el contexto izquierdo de primer nivel; esto se sabe, porque si traducimos las etiquetas a texto tenemos las siguientes combinaciones:

- Coor-sp: y de la República “,” por consiguiente...
- Sp-de: de la República “,” por consiguiente...
- Sn: la República “,” por consiguiente...
- Grup-nom-fs y w-fs: República “,” por consiguiente...

Los niveles contextuales consisten, a partir del nodo padre en común con la palabra a ser analizada; si se analizara el segundo nivel izquierdo, empezaría por la etiqueta “DA0FS0” hasta recorrer el árbol llegando al nodo padre, sin considerar este último.

Sobre este enfoque se desarrolló un método automático que, dada la forma de la palabra, el lema o la etiqueta Eagle, y el árbol sintáctico de Freeling; proporciona como

resultado una tabla de estadísticas, donde se observan los patrones contextuales de las palabras analizadas.

Por otra parte, con el apoyo de ETE Toolkit[13] se desarrolló el método de extracción automática de patrones. ETE Toolkit es un marco en python que permite el análisis y visualización de árboles. La manipulación de esta herramienta con los árboles hace más fácil interactuar con los árboles de Freeling. ETE Toolkit se desarrolló en un principio para la filogenómica, por lo que utilizarlo en árboles de Freeling tendría mucho potencial, por consiguiente podría utilizarse para muchos análisis más.

3.2. Método para la búsqueda extendida con expresiones regulares de patrones morfosintácticos

Las expresiones regulares, también conocidas como regex, regexp o regexen, son patrones textuales que realizan búsquedas a nivel de sintaxis[14].

Sin embargo en este trabajo se extendió su funcionalidad, a tal grado de poder realizar búsquedas a nivel morfosintáctico. Para llegar a esa funcionalidad se realiza un pre-procesamiento de análisis morfosintáctico.

Se procede de la siguiente manera:

1. Se proporciona manualmente un texto de entrada, y se analiza automáticamente el texto extrayendo el contexto de las palabras analizadas.
2. De manera automática, se obtiene un árbol con la API de Freling y se codifica al formato Newick para su posterior utilización con ETE.
3. Se eliminan etiquetas redundantes automáticamente.
4. Se crea automáticamente una expresión regular con la posibilidad de respetar la posición de las etiquetas Eagle o Chunk.
5. Se proporciona manualmente otro texto a analizar para encontrar marcadores argumentativos.
6. Se busca la representación textual de las etiquetas en el texto proporcionado de manera automática.
7. Se sustituye automáticamente la representación textual a cada una de las etiquetas que están en la expresión regular.

En el paso 6 se aplica la expresión regular al segundo texto proporcionado, para encontrar los marcadores que respeten el patrón extraído del primer texto.

El paso 4 puede omitirse, entonces se debe colocar una expresión regular propia o la que genera el sistema, por ejemplo considerando las etiquetas con la mayor probabilidad, esto lo proporciona el método anterior (sección 3.1).

4. Experimentos y resultados

Dado que Freeling maneja una precisión aproximada del 97%[15], se eligió para realizar el etiquetado POS. De igual manera, cabe señalar que la precisión que arrojan los métodos desarrollados dependerá en parte de Freeling.

Para analizar la efectividad de los métodos, se realizó una consulta manual de 200 discursos de la presidencia de la República y secretarías federales, de los cuales 29 contienen marcadores conocidos y de ellos se extrajeron manualmente varios

argumentos con diferentes marcadores de conclusión sugeridos por Capaldi. En la siguiente tabla (tabla 2) se indican los marcadores extraídos:

Tabla 2. Marcadores argumentativos identificados en discursos políticos.

	Marc. Arg.	No Marc. Arg.
<i>Como resultado</i>	3	3
<i>De ahí</i>	5	1
<i>En consecuencia</i>	9	
<i>Implica</i>	1	
<i>Se llegó</i>	1	
<i>Por consiguiente</i>	1	
<i>Por lo tanto</i>	17	2
<i>Así</i>	9	37
<i>Por ello</i>	6	1
Total	52	44

La intención es generar expresiones regulares que incluyan los patrones antes indicados. De la tabla anterior se observa que los más utilizados son “por lo tanto”, “en consecuencia” y “así”. Dado que este último presenta mucha ambigüedad porque puede funcionar como adverbio, conjunción, interjección, entre otras; se decidió tomar los dos primeros para hacer la generación automática de expresiones regulares. En total se logró generar automáticamente 16 expresiones regulares (16 pruebas), de las cuales fueron dos las que lograron identificar la mayor cantidad de palabras que sí cumplen la función de marcadores.

El siguiente fragmento es tomado de una expresión regular. Fue generada por el sistema, tomando como datos de entrada 20 argumentos con el marcador “por lo tanto” y 5 de “en consecuencia” obtenidos de distintos discursos:

```

\s((?:RG_)(?:SPS00_)\s[\wáéíóú]+(?:\s(?:[\wáéíóú]+))?(?:CS_)(?:\s(?:[\wáéíóú]+))?(?:P00CN000_)\s[\wáéíóú]+(?:\s(?:[\wáéíóú]+))?(?:Fc_)(?:P00C_)(?:VMIP_)(?:DA0F_)(?:Fc_)(?:P00C_)(?:VMIF_)(?:VMN0_)|(?:DA0F_)(?:NCFS_)(?:SPS0_)(?:CS_...)[^\w]
    
```

La expresión regular mostrada contiene etiquetas POS truncadas para considerar una mayor posibilidad de encuentros, ya que además de que se extrajeron únicamente los patrones POS también se consideró la posición, por ejemplo: el contexto de un argumento en particular con el marcador “por lo tanto” contiene las etiquetas PP1,VMI,Fc; y se está respetando la posición de estos. Es decir, primero tiene que estar la etiqueta PP1, seguida de la etiqueta VMI y posteriormente la etiqueta Fc, así también con los demás argumentos analizados.

Un ejemplo de la expresión regular mostrada es la identificación del marcador de conclusión “de ahí” del siguiente argumento: “El diagnóstico de la pobreza y el diálogo de la pobreza se vuelve cada vez más complicado y de ahí la importancia en que permanentemente midamos para ver cómo estamos evolucionando”[16]. El contexto

derecho del marcador “de ahí” es: “la importancia en que” que en términos morfosintácticos es: DA0F → la, NCFS → importancia, SPS0 → en, CS → que. Así la expresión regular identificó un nuevo marcador tomando como base los marcadores “por lo tanto” y “en consecuencia”

Tabla 3. Resultados de los marcadores argumentativos encontrados con el método de inducción.

Inducción de patrones (Reglas-aplicadas)	Precisión	Cobertura
Experimento 1	56%	25%
Experimento 2	92%	25%

Se realizaron algunos experimentos, de las cuales dos de ellas fueron las más precisas, los resultados se pueden observar en la tabla 3. El experimento uno tiene las siguientes características: posición de los patrones, que se refiere al orden de las etiquetas POS; contexto izquierdo y derecho (3 niveles), se refiere a que sólo se tomaron 3 palabras a la izquierda y derecha; patrones de “por lo tanto” y “en consecuencia”, son los marcadores analizados; y truncado a 2, se refiere a que no se consideró toda la anotación morfosintáctica. El experimento dos, sólo se consideró el contexto derecho (4 niveles); es decir cuatro palabras a la derecha; truncado a cuatro, es decir los cuatro primeros caracteres de las etiquetas; posición de los patrones, el orden de las etiquetas fueron tomados en cuenta; tipo de marcador, se refiere al tipo de información morfológica, por ejemplo, marcadores de tipo adverbio; y patrones de “por lo tanto” y “en consecuencia”, que son los marcadores analizados.

5. Conclusiones y trabajos futuros

Respecto a los resultados que arrojaron baja precisión, posiblemente se deba a que no se consideraron más argumentos de prueba para extraer sus patrones y de igual forma, es posible que hiciera falta analizar otros tipos de marcadores para luego extraer sus patrones contextuales. Sin embargo, debido a que no se encontró ningún corpus de argumentación en español se tuvo que identificar manualmente argumentos de varios discursos políticos, y desde luego la dificultad que existe al identificar argumentos con algunos marcadores ambiguos hizo el trabajo más tedioso.

Por otro lado, sobre la explicación anterior, se plantea la posibilidad de aplicar las ideas propuestas para tener una variedad más rica de patrones, para que así, encuentre otro tipo de marcadores que cumplan con los requisitos del patrón POS. De tal forma, que aplicando este principio se incremente la precisión y cobertura.

Los métodos desarrollados serán optimizados para aplicarse en diferentes escenarios. Asimismo, la posibilidad de eliminar redundancia de información al encontrar los patrones y las palabras sustituidas.

Además, se realizarán más pruebas que nos permitan identificar más marcadores de conclusión y empezar a realizar pruebas con las premisas.

Los métodos desarrollados pueden ser utilizados para contextos de diferentes palabras. Por ejemplo, es posible ser utilizado para encontrar palabras claves que indiquen definiciones[10].

De igual forma, para la expresión regular mencionada al principio (Figura 1), se plantea la posibilidad de evitar ambigüedad en los argumentos sobre determinados marcadores, por ejemplo “así”. Según el diccionario de la RAE, la palabra “así” puede tener muchos significados dependiendo de su contexto, por ejemplo: “con sueldos así no se puede vivir”, en esta oración “así” está funcionando como adjetivo, y sus sinónimos pueden ser las palabras “semejante o similar”[17], claramente se aprecia que esta palabra no actúa como marcador argumentativo. Por ello, los marcadores ambiguos pueden ser descartados de la identificación teniendo mayor cantidad de ejemplos para diferenciarlos de los que efectivamente son marcadores de argumentos.

Agradecimiento. Los autores agradecen al Consejo Nacional de Ciencia y Tecnología por el apoyo otorgado al presente trabajo.

Referencias

1. Weston, A.: Las claves de la argumentación. 4th ed. Grupo Planeta (2011)
2. Padró, L.: Analizadores multilingües en freeling. *Linguamática*, Vol. 3, No. 2, pp. 13–20 (2012)
3. Ferrari, E.: Los argumentos políticos, [Online]. Available: http://www.elnortedecastilla.es/20081219/articulos_opinion/argumentos-politicos-20081219.html. [Accessed: 04-May-2016] (2008)
4. Educación: Cómo hacer un discurso perfecto, según el escritor que se los redacta a Obama. [Online]. Available: http://www.elconfidencial.com/alma-corazon-vida/2015-01-15/6-consejos-para-elaborar-un-discurso-convincente-segun-el-asesor-de-obama_621098/. [Accessed: 04-May-2016].
5. Capaldi, N.: Como ganar una discusión. Tercera (2011)
6. Avallone, J.: Regexper [Online]. Available: www.regexper.com.
7. Microsoft: Retroceso en expresiones regulares. [Online]. Available: [https://msdn.microsoft.com/es-es/library/dsy130b4\(v=vs.110\).aspx#controlling_backtracking](https://msdn.microsoft.com/es-es/library/dsy130b4(v=vs.110).aspx#controlling_backtracking) [Accessed: 20-Mar-2016]
8. Calsamiglia, H., Tusón, A.: Las cosas del decir: Manual del análisis del discurso (1999)
9. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. *Proceedings of the 14th conference on Computational linguistics*, Vol. 2, pp. 539–545 (1992)
10. Alarcón, R., Sierra, G., Bach, C.: Developing a Definitional Knowledge Extraction System. *Proc. 3rd Lang. Technol. Conf.*, pp. 374–379 (2007)
11. Lloberes M., Padro, L.: Freeling shallow parser: Chunking tags for spanish [Online]. Available: <http://devel.cpl.upc.edu/freeling/svn/trunk/doc/grammars/esCHUNKtags>. [Accessed: 03-Mar-2016]
12. P. de la República: Estado Mayor Presidencial | Presidencia de la República | Gobierno | gob.mx. [Online]. Available: <http://www.gob.mx/presidencia/acciones-y-programas/estado-mayor-presidencial-14579>. [Accessed: 03-Mar-2016] (2015)
13. Huerta-Cepas, J., Serra, F., Bork, P.: ETE 3: Reconstruction, analysis and visualization of phylogenomic data. *Mol. Biol. Evol.*, p. msw046 (2016)
14. López, F., Romero, V.: *Mastering Python Regular Expressions*. Packt Publishing Ltd, (2014)
15. Padró, L., Collado, M., Reese, S., Lloberes, M., Castellón, I.: *FreeLing 2.1: Five Years of Open-Source Language Processing Tools*(2010)
16. S. de D. Social: Mensaje del secretario Meade en la reunión del Consejo Directivo de la Cámara Nacional de la Industria de la Transformación (Canacintra). [Online]. Available: <http://www.gob.mx/sedesol/prensa/mensaje-del-secretario-meade-en-la-reunion-del->

Felipe Ojeda-Cruz, Noé Alejandro Castro-Sánchez, Héctor Jiménez-Salazar

consejo-directivo-de-la-camara-nacional-de-la-industria-de-la-transformacion-canacinfra.
[Accessed: 06-May-2016] (2015)

17. DLE: así - Diccionario de la lengua española - Edición del Tricentenario. [Online]. Available: <http://dle.rae.es/?id=3zT41MT>. [Accessed: 14-Mar-2016]

Modelo difuso para evaluar la pertinencia de un programa educativo de nivel superior

Sandra Silvia Roblero Aguilar^{1,2}, Héctor Rafael Orozco Aguirre²,
Saturnino Job Morales Escobar²

¹ Instituto Tecnológico de Tlalnepantla, Estado de México,
México

² Centro Universitario UAEM Valle de México, Estado de México,
México

ssrauaemex@hotmail.com, hrorozcoa@uaemex.mx, sjob_d@yahoo.com

Resumen. Este artículo tiene como finalidad presentar un modelo, el cual utiliza un sistema de inferencia difuso para evaluar la pertinencia de un programa educativo de nivel superior con respecto al Examen General de Egreso de Licenciatura (EGEL) del Centro Nacional de Evaluación para la Educación Superior (CENEVAL). Los siguientes factores se consideraron para desarrollar el modelo: impacto de cada Unidad de Aprendizaje (UDA) sobre las áreas que evalúa el EGEL, calificación del alumno en cada UDA, apreciación estudiantil promedio en cada UDA, pérdida de conocimiento por años de egreso y ganancia de conocimiento por años de experiencia profesional.

Palabras clave: Modelo difuso, pertinencia, EGEL, programa educativo.

Fuzzy Model to Evaluate the Relevance of a Higher Educational Program

Abstract. This article presents a model, which uses a fuzzy inference system to evaluate the relevance of a higher educational program in regard to the General Examination for the Degree Exit (EGEL) of the National Center for Higher Education Assessment (CENEVAL). To develop the model the following factors were considered: the impact of each Learning Unit (UDA) on areas EGEL assessment, the score obtained by the student in each UDA, the average student appreciation for each UDA, the loss of knowledge from years of graduation, and the gain of knowledge from years of professional experience.

Keywords: Fuzzy model, relevance, EGEL, educational program.

1. Introducción

En México, el Programa Sectorial de Educación 2013-2018, busca lograr como uno de sus objetivos “Fortalecer la calidad y pertinencia de la educación media superior, superior y formación para el trabajo, a fin que contribuyan al desarrollo de México” [1]. Ahora bien, para lograr el aseguramiento de la calidad de los programas y la fortaleza de las Instituciones de Educación Superior (IES), en nuestro país se han impulsado mecanismos, de los cuales sobresalen: las evaluaciones que llevan a cabo los Comités Interinstitucionales para la Evaluación de la Educación Superior (CIEES); las acreditaciones de programas que se efectúan al amparo del Consejo para la Acreditación de la Educación Superior (COPAES); y los Exámenes Generales de Egreso de la Licenciatura (EGEL) del Centro Nacional de Evaluación (CENEVAL).

Dado lo anterior, resulta necesario hacer una evaluación de la relación de los Programas Educativos (PE) con respecto a las áreas que evalúa el EGEL. Tomando como ejemplo al Centro Universitario UAEM Valle de México (CU UAEM VM), perteneciente a la Universidad Autónoma del Estado de México (UAEM), desde 2003 se ha impulsado al EGEL como una modalidad de titulación. Aunque los resultados no han sido los esperados, como se observará más adelante, es conveniente decir que tampoco se han realizado estudios comparativos de los contenidos de los PE con relación a las áreas que evalúa el EGEL. En esta investigación se muestra un modelo basado en la teoría de la lógica difusa y su aplicación, en un análisis de la relación de los contenidos de cualquier PE con respecto al EGEL que le corresponda. Para su validación se tomará como caso de estudio al examen en Ingeniería Computacional (EGEL-ICOMPU) aplicado al PE Ingeniería en Sistemas y Comunicaciones (ISC) versión flexible [2] del CU UAEM VM, que entró en vigor en 2004. El trabajo aquí presentado, se apega a lo establecido en el paradigma cuantitativo, que de acuerdo con Tamayo [3], será mostrado como una investigación de tipo descriptiva.

2. Descripción del EGEL CENEVAL

Las calificaciones consideradas en el EGEL-CENEVAL, están expresadas en una escala especial llamada Índice CENEVAL (IC) [4] que va de 700 puntos, calificación más baja; a 1300 puntos, calificación más alta.

En cada una de las áreas evaluadas por el EGEL, se consideran los siguientes tres niveles de desempeño [4]: Aún No Satisfactorio (ANS), Satisfactorio (DS) y Sobresaliente (DSS), en los cuales se clasifica a los sustentantes en función del Testimonio de Desempeño (TD) mostrado, esto es, no satisfactorio si obtiene menos de 1000 puntos, Testimonio de Desempeño Satisfactorio (TDS) si está en el rango de 1000 a 1149 puntos, y Testimonio de Desempeño Sobresaliente (TDSS) en el rango de 1150 a 1300 puntos del IC.

2.1. Trabajos relacionados con el EGEL

A continuación, se citan y describen algunos trabajos:

- Aspectos del EGEL y los resultados de egresados que lo aplicaron durante los años 2001 y 2007 de la carrera de Ingeniería Química del Instituto Tecnológico de Aguascalientes [5].
- Relación entre el promedio académico que obtienen los alumnos de contaduría pública de la Unidad Académica Multidisciplinaria Zona Media, dependiente de la Universidad Autónoma de San Luis Potosí, con el puntaje que logran en el EGEL [6].
- Estadísticas de los resultados que se han obtenido en el EGEL para los egresados de la Universidad de Colima y compartir su experiencia en la elaboración de un examen que se ha aplicado a ocho generaciones de egreso [7].
- Resultados del EGEL de los egresados de los años 2008 a 2010 de la licenciatura en Ciencias de la Educación de la Facultad de Ciencias Humana de la Universidad Autónoma de Baja California [8].
- Mecanismos de mejora de la calidad de los programas de licenciatura que oferta el Instituto Tecnológico de Pachuca, además de presentar estrategias y acciones realizadas en materia de evaluación por medio del EGEL [9].

3. Lógica difusa

La lógica difusa está relacionada y fundamentada en la teoría de los conjuntos difusos, en la cual, el grado de pertenencia de un elemento a un conjunto está determinado por una función de pertenencia que puede tomar valores reales comprendidos en el intervalo $[0,1]$, [10], [11], [12]. De esta manera, mientras que en el marco de la lógica clásica un PE tiene pertinencia o no con las áreas que evalúa el EGEL, dándole un valor de 1 si es pertinente y 0 en caso contrario. En la lógica difusa, se obtiene un nivel de cumplimiento de la pertinencia, entre más cercano a cero, será menos pertinente y cuando sea más cercano a 1 será más pertinente.



Fig. 1. Sistema de inferencia difuso.

Las etapas que deben cumplirse para el montaje de un sistema de inferencia difuso [12], [13] se muestran en la Fig. 1. Estas se explican a continuación:

- Etapa 1: se basa en un proceso donde las variables tienen un grado de incertidumbre metalingüístico. Es decir, el rango de valores de cada variable puede clasificarse por conjuntos difusos, originando el universo del discurso. Con ello, los valores pasan a un proceso de fusificación que los categoriza en un rango de pertenencia entre 0 y 1 a un conjunto difuso.
- Etapa 2: se proponen reglas lingüísticas conocidas como de inferencia. Con esto, el grado de pertenencia de cada una de las variables se evalúa en un subconjunto de estas reglas.

- Etapa 3: consiste en determinar los valores óptimos de salida, mediante un mecanismo conocido como defusificación, el cual consiste en pasar el grado de pertenencia, proveniente de la consecuencia de la regla de inferencia activada, a un valor nítido o real, con el fin de obtener un valor cuantificable.

4. Modelo difuso

4.1. Propuesta

El esquema general del modelo difuso que se propone se puede visualizar en la Fig 2, donde se consideran dos entradas, la primera: son las UDAs de los PEs, a cada una de ellas se les asigna una ponderación, la cual es determinada por el coordinador de carrera, con respecto a los contenidos de cada UDA y la relación con las áreas que evalúa el EGEL, esta asignación, representa la relevancia o grado de impacto del conocimiento en cada una de las áreas. Con esta base y tomando en consideración las calificaciones de las UDAs del sustentante durante su trayectoria académica y la apreciación estudiantil (evaluación realizada por el alumno de la actividad del docente en cada UDA), se realiza un cálculo de estimación simple; la segunda entrada es el histórico de resultados del EGEL, este histórico permite calcular el factor de impacto por área, dependiendo de la pérdida o ganancia de conocimiento del sustentante en cada área con base en los años de experiencia laboral y tiempo de egreso, de esta asignación se tiene como resultado la ponderación acumulada por área, con lo anterior se puede determinar el cálculo de estimación por egreso y por experiencia, lo cual pondera el nivel de impacto de los contenidos de las UDAs por área.

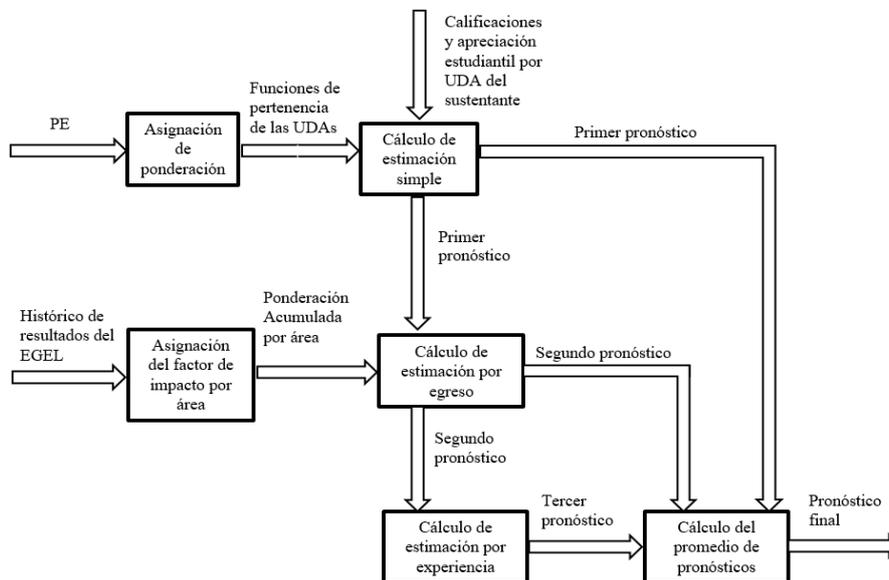


Fig. 2. Esquema general del modelo difuso.

4.2. Análisis del PE

El modelo propuesto consta de lo siguiente:

A) El conjunto de UDAs de un PE, que a partir de este momento será el universo del discurso, está determinado de manera extensional por:

$$\mu = \{UDA_1, UDA_2, UDA_3, \dots, UDA_n\}. \quad (1)$$

A cada **UDA** se le asigna una ponderación con respecto a sus contenidos y la relación con cada una de las áreas del EGEL, que se puede definir como: $P =$ “tener membresía para cada una de las áreas correspondientes del EGEL”. Se define P como el subconjunto de UDAs con membresía a dichas áreas, esta ponderación representa la relevancia o grado de impacto del conocimiento en cada una de las áreas que evalúa el EGEL.

Al finalizar de ponderar a cada UDA con respecto a lo que evalúa el EGEL, se aplica la siguiente ecuación para cada área:

$$VPA = \Sigma(PUDA), \quad (2)$$

donde **VPA**: valor de ponderación acumulada por área,

PUDA: ponderación asignada a cada UDA en esa área.

El resultado de la aplicación de esta ecuación proporciona la ponderación acumulada por área.

Por otra parte, inicialmente se consideraron 2 factores de impacto, que determinan si un sustentante alcanza o no un TD en el EGEL, estos factores son:

1. La calificación obtenida por el sustentante en cada una de la UDAs en su trayectoria académica.
2. La calificación proporcionada por el alumno a los docentes de cada UDA en el proceso de evaluación docente.

Con los dos primeros factores se obtiene la siguiente estimación (representa el primer pronóstico del esquema general del modelo difuso):

- Estimación Simple por Área (**ESA**): en esta se considera el valor máximo de **VPA** y los rangos de puntuación del IC, tal y como se muestra en la siguiente ecuación:

$$ESA = \frac{PPA}{\text{Max}(VPA)} * 6 + 700, \quad (3)$$

donde **PPA** es el Puntaje de Ponderación por Área, la cual se obtiene de la siguiente ecuación:

$$PPA = \Sigma((CUDA * F1 + EDO * F2) * PUDA), \quad (4)$$

donde:

CUDA es la calificación de la UDA.

EDO es la evaluación promedio del docente asignado por los estudiantes en esa UDA.

F1 es el factor de impacto de la calificación de la UDA, expresado en porcentaje.

F2 es el factor de impacto de la calificación del docente, expresado en porcentaje.

Los factores de impacto son determinados heurísticamente por el coordinador del PE.

B) Con base en los resultados obtenidos por los sustentantes, se realizó un ajuste a posteriori en el modelo y se agregan dos nuevos factores de impacto: el tiempo de egreso y los años de experiencia profesional, con lo anterior se generan dos estimaciones más, descritos como sigue:

- Estimación Por años de Egreso (**EPE**): implica el conocimiento que el sustentante ha perdido por el tiempo que tiene de egresado, este factor, definido como Factor de Pérdida (**FP**), debe de calcularse tomando en cuenta tres grados de pérdida, determinados como alto, medio y bajo. Estos grados están relacionados con los puntajes mínimos requeridos esperados en la ecuación 3.

Tabla 1. Cálculo del **FP** por años de egreso.

Grado de pérdida	Valor mínimo requerido	Puntaje mínimo requerido en ESA	Tiempo de egreso	TD esperado
FP _{Alto}	150 * Max (VPA) 6	850	> 2 años	ANS
FP _{Medio}	300 * Max (VPA) 6	1000	>= 1 y <= 2 años	TDS
FP _{Bajo}	375 * Max (VPA)	1075	< 1 año	TDS

El **FP** está asociado al tiempo de egreso del sustentante y al puntaje mínimo requerido en la Estimación Simple por Área.

Al aplicar el **FP** se puede obtener la siguiente ecuación **EPE**:

$$EPE = \frac{(PPA * (\frac{FP}{VPA * 100}))}{Max (VPA)} * 6 + 700. \quad (5)$$

- Estimación por Años de Experiencia (**EAE**): la que el egresado tenga en alguna de las áreas evaluadas por el EGEL. En este factor, se toma en cuenta la experiencia que el sustentante pudiese haber obtenido en cada una de las áreas del EGEL, en función de la o las actividades desempeñadas en el ámbito profesional. Para este factor se considera el resultado de **VPA** obtenido en cada una de las áreas del EGEL y el **FP**, es decir, el valor mínimo requerido definidos en la Tabla 1, este factor también se da en tres grados: alto, medio y bajo. Con lo anterior se obtiene la Tabla 2.

Tabla 2. Cálculo del factor de ganancia.

Grado de ganancia	Ganancia	Tiempo de experiencia	TD esperado
FG_{Alto}	$VPA * 100$ de cada área FP_{Bajo}	> 2 años	TDSS
FG_{Medio}	$VPA * 100$ de cada área FP_{Medio}	≥ 1 y ≤ 2 años	TDS
FG_{Bajo}	$VPA * 100$ de cada área FP_{Alto}	< 1 año	ANS

Al aplicar este Factor de Ganancia (FG) por años de experiencia, se tiene como resultado la siguiente expresión:

$$EAE = \frac{PPA + ((VPA * 100) * FG)}{\text{Max}(VPA)} * 6 + 700. \quad (6)$$

Considerando lo anterior, el siguiente paso es calcular el promedio de las tres estimaciones y con ello se obtiene la siguiente ecuación:

$$PAE = \frac{ESA + EPE + EAE}{3}, \quad (7)$$

donde PAE es el pronóstico del área del EGEL.

La ecuación 7 se debe realizar para cada una de las áreas del EGEL.

4.3. Definición de las reglas difusas

Fuzzy Control Language (FCL) [14], es el lenguaje para definir las reglas del sistema de inferencia difuso que se utiliza en las clases de Java del paquete jFuzzyLogic. FCL es un estándar incluido en el IEC (International Electrotechnical Commission).

Para poder escribir las reglas difusas se consideraron las áreas que evalúa el EGEL y los criterios establecidos para el otorgamiento del TD.

Considerando que no importa el orden de las áreas, se aplica la fórmula de combinación, con la finalidad de establecer el número de reglas requeridas.

$$C_k^n = \frac{n!}{k!(n-k)!}, \quad (8)$$

donde:

n es número de áreas que se evalúan para un PE.

k variables de entrada (número de criterios establecidos para el otorgamiento del TD).

Las reglas del sistema de inferencia difuso, definidas para obtener el TD tienen la forma siguiente:

- Reglas difusas para TDSS
- $RULE_j$: IF (área_i IS ds OR área_i IS dss) AND
- (área_{i+1} IS ds OR área_{i+1} IS dss) AND
- ...
- (área_{i+n-1} IS ds AND área_{i+n-1} IS dss) THEN
- td IS dss;
- Reglas difusas para TDS
- $RULE_{j+k}$: IF (área_i IS ds OR área_i IS dss) AND
- o (área_{i+1} IS ds OR área_{i+1} IS dss) AND
- ...
- o (área_{i+n-1} IS ds OR área_{i+n-1} IS dss) THEN
- td IS ds;
- Reglas difusas para ANS
- $RULE_{j+k+1}$: IF (área_i IS ans AND área_{i+1} IS ans) THEN
- td IS ans;

Se definen $RULE_j$, para $1 \leq j \leq m$ y área_i, para $1 \leq i \leq n$.

Para $1 \leq k \leq 4|6|10$.

Para $1 \leq l \leq 6|10$.

Donde m corresponde al número total de reglas formadas para un PE y n el número de áreas que se evalúan para un PE.

4.4. Fusificación y defusificación

La fusificación y defusificación se aplica a cada una de las áreas de evaluación, considerando la distribución del IC.

```
FUZZIFY áreai
  TERM ans := (0.5384, 1) (0.7669, 1) (0.7700, 0);
  TERM ds := (0.7676, 0) (0.7692, 1) (0.8823, 1) (0.8853, 0);
  TERM dss := (0.8830, 0) (0.8846, 1) (1, 1);
END_FUZZIFY

DEFUZZIFY td
  TERM ans := (0.5384, 1) (0.7669, 1) (0.7700, 0);
  TERM ds := (0.7676, 0) (0.7692, 1) (0.8823, 1) (0.8853, 0);
  TERM dss := (0.8830, 0) (0.8846, 1) (1, 1);
  METHOD : COG;
  DEFAULT := 0.5384;
  RANGE := (0.5384 .. 1);
END_DEFUZZIFY
```

En la Tabla 3, se muestra el mapeo del IC con respecto a los niveles de desempeño del EGEL para la defusificación, la cual se obtiene aplicando una regla de tres, donde 1300 es el 100%.

Table 3. Mapeo del IC para la defusificación.

Niveles de Desempeño	Puntos del IC/valor de mapeo			
	ANS	700/0.5384	997/0.7669	1001/0.7700
DS	998/0.7676	1000/0.7692	1147/0.8823	1151/0.8853
DSS	1148/0.8830	1150/0.8846	1300/1	

Es importante mencionar que para la defusificación se usa el centro de gravedad (COG), lo cual implica que el valor a obtener para el TD se ubica en el centro del rango de pertenencia.

5. Caso de estudio

5.1. Descripción del EGEL-ICOMPU y sus resultados en el PE DE ISC

En el EGEL-ICOMPU [4], aplicado a egresados de ISC, para que el sustentante obtenga un TDS: de las cinco áreas a evaluar, al menos debe tener cuatro áreas con DS o DSS y para obtener el TDSS: al menos en dos debe alcanzar DSS y las restantes con DS. Las áreas del EGEL-ICOMPU son las siguientes:

- Selección de Sistemas Computacionales para Aplicaciones Específicas (SSCAE).
- Nuevas Tecnologías para la implementación de Sistemas de Cómputo (NTISC).
- Desarrollo de Hardware y Software Asociado para Aplicaciones Específicas (DHSAE).
- Adaptación de Hardware y/o Software para Aplicaciones Específicas (AHSAE).
- Redes de Cómputo para Necesidades Específicas (RCNE).

Al hacer un análisis de los resultados de sustentantes del EGEL-ICOMPU a nivel nacional, CU UAEM VM y PE ISC entre 2013 y 2015, se tiene la Tabla 4.

Tabla 4. Comparativo de resultados del EGEL-ICOMPU.

	ANS	TDS/TDSS
Nacional	50%	50%
CU UAEM VM	46%	54%
ISC	54%	46%

Como se puede observar en la Tabla 4, a nivel nacional el 50% de los sustentantes logran obtener un TD aprobatorio (TDS o TDSS), mientras que el mismo porcentaje no lo obtiene (ANS). En el caso del CU UAEM VM, el 54% de sus sustentantes aprueba y para la carrera de ISC sólo el 46% logra este resultado. Lo anterior, requiere de una atención inmediata para que se puedan establecer mecanismos que conlleven a que un

mayor número de sustentantes logren alcanzar un TD, aunado a esto, si se considera que en el CU UAEM VM, el EGEL es una de las modalidades de titulación, sus resultados impactan directamente en el índice de titulación.

Considerando que el PE de ISC consta de 53 UDAs, se tiene como resultado la Tabla 5, en donde se muestra la distribución de las ponderaciones de cada UDA por área.

Al finalizar se pondera a cada UDA con respecto a lo que evalúa el EGEL, se aplica la ecuación 2.

En el resultado de la aplicación, se observa cuáles de las UDAs tienen mayor impacto en el EGEL-ICOMPU con respecto al PE de ISC, en la Tabla 6, se muestra el *VPA* y el porcentaje que representa.

Aplicando las ecuaciones 4 y 5, se tienen como valores: 275 para un FP_{Alto} , 550 para un PF_{Medio} y 687.5 para un FP_{Bajo} . Es importante mencionar que los valores determinados por el coordinador de ISC para las variables $F1$ y $F2$ fueron 90 y 10% respectivamente, lo anterior indica que la calificación obtenida en cada UDA tiene mayor impacto en el resultado en comparación a la apreciación estudiantil.

Tabla 5. Distribución de ponderación con respecto a las áreas del EGEL-ICOMP.

No.	NOMBRE UDA	SSCAE	NTISC	DHSAE	AHSAE	RCNE
1	ALGEBRA LINEAL	0.2	0.2	0.2	0.2	0.2
2	ALGEBRA Y GEOMETRIA ANALITICA	0.2	0.2	0.2	0.2	0.2
3	CALCULO DIFERENCIAL E INTEGRAL	0.2	0.2	0.2	0.2	0.2
4	CALCULO VECTORIAL	0.2	0.2	0.2	0.2	0.2
5	ECUACIONES DIFERENCIALES	0.2	0.2	0.2	0.2	0.2
6	ELECTROMAGNETISMO	0.2	0.2	0.2	0.2	0.2
7	ESTATICA Y DINAMICA	0.2	0.2	0.2	0.2	0.2
8	INGLES C1	0.2	0.2	0.2	0.2	0.2
9	INGLES C2	0.2	0.2	0.2	0.2	0.2
10	INTRODUCCION A LA COMPUTACION	0.2	0.2	0.2	0.2	0.2
11	PROBABILIDAD Y ESTADISTICA PARA INGENIEROS	0.2	0.2	0.2	0.2	0.2
12	QUIMICA	0.2	0.2	0.2	0.2	0.2
13	ADMINISTRACION	0.2	0.2	0.2	0.2	0.2
14	CONTABILIDAD	0.2	0.2	0.2	0.2	0.2
15	ECOLOGIA, ETICA Y NORMATIVIDAD	0.2	0.2	0.2	0.2	0.2
16	TECNICAS DE COMUNICACION	0.2	0.2	0.2	0.2	0.2
17	ALGORITMOS Y ESTRUCTURA DE DATOS	0.4	0.3	0.1	0.2	0
18	ARQUITECTURA DE COMPUTADORAS	0.3	0.2	0.3	0.2	0
19	BASES DE DATOS	0.5	0.3	0	0.2	0
20	CIRCUITOS ELECTRICOS	0.1	0.1	0.4	0	0.4
21	DESARROLLO DE PROYECTOS	0.1	0.3	0.1	0.3	0.2
22	ELECTRONICA ANALOGICA	0.1	0.1	0.4	0	0.4
23	FORMULACION Y EVALUACION DE PROYECTOS	0.3	0.2	0.1	0.3	0.1
24	FUNDAMENTOS DE BASES DE DATOS	0.4	0.3	0	0.3	0
25	FUNDAMENTOS DE PROGRAMACION	0.3	0.3	0.1	0.3	0
26	INGENIERIA DE SOFTWARE	0.4	0.5	0	0.1	0
27	INTRODUCCION A LA INGENIERIA	0.2	0.2	0.2	0.2	0.2
28	INVESTIGACION DE OPERACIONES	0.2	0.2	0.2	0.2	0.2
29	LENGUAJES DE BAJO NIVEL	0	0.3	0.3	0.2	0.2
30	LENGUAJES FORMALES Y AUTOMATAS	0.2	0.2	0.2	0.2	0.2
31	LOGICA MATEMATICA	0.2	0.2	0.2	0.2	0.2
32	MATEMATICAS DISCRETAS	0.2	0.2	0.2	0.2	0.2
33	METODOLOGIA DE LA INVESTIGACION	0.2	0.2	0.2	0.2	0.2
34	METODOS NUMERICOS	0.2	0.2	0.2	0.2	0.2
35	PROGRAMACION ORIENTADA A OBJETOS	0.2	0.3	0.2	0.3	0
36	PROTOSCOLOS DE COMUNICACION DE DATOS	0.1	0.1	0.1	0.3	0.4
37	REDES	0.1	0.1	0.1	0.1	0.6
38	SISTEMAS DE INFORMACION	0.4	0.3	0.1	0.1	0.1
39	SISTEMAS DIGITALES	0.2	0.2	0.2	0.2	0.2
40	SISTEMAS OPERATIVOS	0	0.3	0.3	0.3	0.1
41	SISTEMAS OPERATIVOS PARA RED	0.1	0.1	0.1	0.4	0.3
42	TEMAS SELECTOS DE SISTEMAS	0.2	0.2	0.2	0.2	0.2
43	ADMINISTRACION DE CENTROS DE COMPUTO	0.2	0.2	0.2	0.2	0.2
44	INTERCONEXION Y SEGURIDAD EN REDES	0.1	0.1	0.1	0.1	0.6
45	RESIDENCIAS PROFESIONALES	0.2	0.2	0.2	0.2	0.2
46	INTELIGENCIA ARTIFICIAL	0.3	0.3	0.1	0.2	0.1
47	SEMINARIO DE TITULACION	0.2	0.2	0.2	0.2	0.2
48	PLANEACION ESTRATEGICA	0.2	0.2	0.2	0.2	0.2
49	SISTEMAS DE INSTRUMENTACION Y CONTROL	0.1	0.1	0.3	0.2	0.3
50	SISTEMAS ELECTRONICOS DE COMUNICACION	0.1	0.1	0.3	0.1	0.4
51	TEORIA DE CONTROL	0.2	0.1	0.3	0.2	0.2
52	TRANSMISION Y COMUNICACION DE DATOS	0.2	0.1	0.1	0.1	0.5
53	SISTEMAS EXPERTOS	0.3	0.3	0.1	0.2	0.1

Tabla 6. Resultado de *VPA* para el PE en ISC.

	Nombre de áreas				
	SSCAE	NTISC	DHSAAE	AHSAE	RCNE
<i>VPA</i>	10.9	11	9.8	10.5	10.8
% <i>VPA</i>	99.09	100	89.09	95.45	98.18

Al aplicar el Factor de Ganancia (*FG*) por años de experiencia, el cual está determinado en la ecuación 6 se tiene como resultado la Tabla 7.

Tabla 7. *FG* para el PE en ISC.

<i>FG</i>	ÁREA EGEL-ICOMPU				
	SSCAE	NTISC	DHSAAE	AHSAE	RCNE
<i>FG</i> _{Alto}	0.6307	0.625	0.7015	0.6547	0.6365
<i>FG</i> _{Medio}	0.5045	0.5	0.5612	0.5238	0.5092
<i>FG</i> _{Bajo}	0.2522	0.25	0.2806	0.2619	0.2546

De acuerdo a la ecuación 8, se generaron 25 reglas difusas, distribuidas como sigue: 10 para el TDSS, 5 para el TDS y 10 para el ANS.

5.2. Análisis de resultados

Entre 2014 y marzo 2015, 26 egresados del PE de ISC plan flexible realizaron el EGEL-ICOMPU, a los cuales se les aplicó el modelo difuso propuesto, al realizar un análisis de los resultados obtenidos, se tiene la Tabla 8, la cual considera el puntaje pronosticado en cada área del EGEL-ICOMPU comparado con el puntaje real obtenido por los sustentantes en dichas áreas.

En la Tabla 8, se observa lo siguiente:

- El área DHSAAE, es la de menor puntaje, tanto en el pronóstico como en el puntaje obtenido, esto tiene sentido, toda vez que en el cálculo del *VPA* se muestra que es el área de menor impacto en el EGEL, es importante realizar acciones que lleven a elevar este puntaje.
- En las cuatro áreas restantes, se observa que en promedio los sustentantes alcanzan más de 1000 puntos, por lo que se alcanzaría un TD, sin embargo, en ninguno de los casos se visualiza un puntaje mayor a 1040, lo que implica, que se corre el riesgo de obtener un ANS.
- En promedio se tiene una aproximación del 96.28%, lo cual representa un buen grado de aproximación.

Tabla 8. Comparativo del puntaje pronosticado vs puntaje obtenido .

No.	Puntaje pronosticado en cada área del EGEL- ICOMPU					Puntaje obtenido por el sustentante en cada área del EGEL-ICOMPU					APROXIMACIÓN
	SSCAE	NTISC	DHSAE	AHSAE	RCNE	SSCAE	NTISC	DHSAE	AHSAE	RCNE	
1	1148	1153	1015	1140	944	1146	1178	1049	1175	915	97.65%
2	918	922	903	917	917	920	947	898	937	829	96.75%
3	924	927	909	920	920	961	885	898	894	851	95.80%
4	930	935	920	930	937	824	947	898	786	915	92.51%
5	932	936	917	928	930	934	957	898	1024	937	97.11%
6	1078	1081	915	929	934	1056	1043	920	916	958	97.95%
7	944	947	1043	1064	941	947	978	1008	1047	958	97.94%
8	1054	932	914	1046	926	1021	957	986	1024	980	95.86%
9	1063	938	923	1054	939	1003	998	898	1047	980	96.08%
10	944	1080	927	940	1075	988	1058	942	980	1001	96.10%
11	1058	935	919	927	1066	1003	957	898	851	1020	95.26%
12	1062	1084	922	933	1061	1074	1149	942	980	1075	97.04%
13	925	929	909	921	1064	947	998	898	980	1094	96.18%
14	1055	932	913	1061	1064	1003	998	964	1137	1113	94.38%
15	1067	928	912	921	1071	1039	988	920	959	1131	96.25%
16	1062	1069	920	1055	1058	1003	1074	898	1069	1020	97.21%
17	1018	1129	1006	1017	1018	1056	1159	1090	1024	1057	96.37%
18	1031	1035	1018	1030	1030	1056	1058	1042	1047	1057	97.86%
19	1025	1029	1010	1022	1021	1021	998	1028	1024	1094	97.67%
20	1057	936	1028	1067	1070	1021	947	1070	1137	1094	96.64%
21	1097	1100	945	1093	1169	1003	1059	942	1024	1113	94.90%
22	1132	1137	1095	1122	1126	1146	1149	1090	1024	1113	97.28%
23	1072	1090	1045	1068	1105	1021	1104	1070	959	1131	95.54%
24	1104	1110	1070	1096	1093	1198	1134	1070	1175	1020	95.23%
25	1125	1130	1090	1115	1117	1198	1134	1049	1114	1150	97.34%
26	1105	1113	1077	1096	1107	1185	1149	1171	1175	1150	94.36%
	1035.77	1020.65	971.73	1015.85	1027.04	1029.77	1038.58	982.19	1019.58	1029.08	96.28%

La evaluación de pertinencia a través del modelo difuso aplicado para el PE en ISC, ha permitido detectar lo siguiente:

- El área de Nuevas Tecnologías para la Implementación de Sistemas de Cómputo, es la de mayor presencia en el EGEL-ICOMPU, esto es, que el sustentante tiene las competencias para seleccionar y aplicar los principios y metodologías de desarrollo necesarios en la generación de nueva tecnología en la implementación de sistemas de cómputo.
- El área de Desarrollo de Hardware y su Software Asociado para Aplicaciones Específicas, es la de menor presencia para el PE ISC, lo que implica que el sustentante no tiene las competencias para analizar distintos problemas y proponer soluciones. Se sugiere fortalecer dicha área, adecuando los contenidos del PE en ISC, para que ello redunde en un mejor aprovechamiento de los alumnos y por ende mayores puntuaciones en el EGEL-ICOMPU.
- Los sustentantes con mayor experiencia en el campo laboral, son lo que han alcanzado un TD.

Dada la generalidad de este modelo, puede ser aplicado a cualquier PE de una IES, para dar respuesta a las siguientes preguntas: ¿Qué áreas de las que evalúa el EGEL tienen mayor o menor impacto en un PE?, ¿Qué grado de impacto tiene cada UDA en un PE con respecto a las áreas del EGEL?, ¿Los factores de tiempo de egreso y experiencia profesional impactan sobre un TD?

6. Conclusiones

La lógica difusa está enfocada en la toma de decisiones cuando existen datos o conocimientos inciertos, existiendo diversas aplicaciones para la vida real y donde se

reemplaza al humano por un sistema difuso basado en reglas. En este trabajo, se eligió esta debido a que ofrece varias ventajas, descritas como sigue:

- Al momento de dar un pronóstico, proporciona una manera sencilla y eficaz para extraer conclusiones de vaguedad, ambigua o información imprecisa. Por lo tanto, simula la toma de decisiones y puede trabajar a partir de datos aproximados para obtener soluciones precisas.
- Incorpora una forma alternativa de pensar, lo que permite que todo pronóstico sea modelado con un nivel de abstracción que refleje conocimiento y experiencia a partir de reglas de inferencia.
- Permite expresar conocimiento con conceptos subjetivos, tales como los resultados que pueden ser obtenidos en el EGEL: ANS, TDS y TDSS, los cuales pueden ser mapeados de manera exacta dentro de los rangos difusos.
- Es un método eficiente que rápidamente proporciona uno o más pronósticos como soluciones.
- Ofrece varios beneficios, tales como el rendimiento, simplicidad, bajo costo y productividad en el modelado de un sistema de pronóstico difuso.

El estudio aquí planteado ayudará, entre otros aspectos, a conocer si existe pertinencia entre los Programas de Estudios de las IES y las áreas que evalúa el EGEL-CENEVAL; los resultados obtenidos, proporcionarán información útil a las autoridades de los espacios académico para tomar decisiones que coadyuven a alcanzar un Testimonio de Desempeño, abordando con ello el problema de la baja Eficiencia de titulación que existe actualmente.

Referencias

1. SEP.: Programa Sectorial de Educación 2013-2018. http://www.sep.gob.mx/work/models/sep1/Resource/4479/4/images/PROGRAMA_SECTORIAL_DE_EDUCACION_2013_2018_WEB.pdf (2015)
2. UAEM.: Curriculum de Ingeniería en Sistemas y Comunicaciones. http://dep.uaemex.mx/curricular/sitio_/index.php (2015)
3. Tamayo, M.: El proceso de la investigación científica. Limusa, pp. 37–45 (2009)
4. CENEVAL.: Guía para el sustentante: Examen General para el Egreso de la Licenciatura en Ingeniería Computacional. http://archivos.ceneval.edu.mx/archivos_portal/20882/GuiadeIEGEL-COMPU.pdf (2015)
5. Trejo, R.: Experiencia de desempeño en el examen EGEL de los egresados de la Carrera de Ingeniería Química del Instituto Tecnológico de Aguascalientes, Conciencia Tecnológica, pp. 41–43 (2008)
6. Izar, J., López, H.: El puntaje que obtienen los estudiantes de contaduría pública en el examen general de egreso de la licenciatura (EGEL) y su relación con el promedio obtenido durante su carrera. *Hitos de Ciencias Económica Administrativa*, pp. 27–34 (2010)
7. Bravo, O., Rodríguez, J.: El Examen General de Egreso de Licenciatura de la Facultad de Lenguas extranjeras: Una experiencia en proceso.

- http://fel.uqroo.mx/adminfile/files/memorias/Articulos_Mem_FONAEL_II/Bravo_Gomez_Omar_&Rodriguez_Reyes_Jose_Miguel.pdf. (2013)
8. Vazquez, J., Bastidas, B., Hiraes, M., Botello, R., Ramírez, A.: La evaluación de egresados de licenciatura, una perspectiva comparada entre los resultados obtenidos en dos versiones del EGEL-CENEVAL. <http://www.saece.org.ar/docs/congreso4/trab45.pdf>. (2015)
 9. León, Y., Palacios, G., Martínez, J., Juárez, F.: Experiencia del Instituto Tecnológico de Pachuca en los procesos de evaluación-EGEL, *Sistemas y Gestión Educativa*, pp. 1016–1021 (2015)
 10. Zadeh, L.: Fuzzy sets, *Information and Control*. pp. 338–353 (1965)
 11. Dubois, D., Prade, H.: *Fuzzy sets and systems: Theory and Applications*. Academic Press (1980)
 12. Kosko, B.: Fuzzy systems as universal approximators. *IEEE*, pp. 1329–1333 (1994)
 13. Mandani, E.: Application of fuzzy logic to approximate reasoning using linguistic synthesis. *IEEE* (1977)
 14. IEC.: International Electrotechnical Commission. [http://www.iec.ch/dyn/www/f?p=103:91:0:::FSP_LANG_ID:25?q=Fuzzy Control Language](http://www.iec.ch/dyn/www/f?p=103:91:0:::FSP_LANG_ID:25?q=Fuzzy%20Control%20Language) (2014)

Algoritmo para generar frases homófonas de calambures desde un enfoque fonético

Roberto Villarejo-Martínez, Noé Alejandro Castro-Sánchez

Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca,
México

{rvillarejo, ncastro}@cenidet.edu.mx

Resumen. Los juegos de palabras son un tipo de humor verbal ampliamente utilizado en todo el mundo. Comúnmente se valen de la homofonía o de frases muy similares en sonido, además de la ambigüedad en los significados de las palabras. En el área de Procesamiento de Lenguaje Natural normalmente esto representa una deficiencia. Sin embargo, en el humor computacional puede verse como una ventaja que puede ser aprovechada en la generación e identificación de estos textos. En este artículo se describen los fenómenos lingüísticos más utilizados en los juegos de palabras. Además se propone un algoritmo para la generación de frases homófonas a partir de una frase, utilizando un enfoque fonético y un lexicón no humorístico ni entrenado para tal propósito. El algoritmo fue capaz de generar 22 de 45 calambures extraídos principalmente de adivinanzas populares.

Palabras clave: calambur, humor computacional, homofonía.

Algorithm for Generate Homophone Phrases of Puns from a Phonetic Approach

Abstract. Wordplays are a type of verbal humor widely used around the world. Usually, they use homophony or phrases very similar in sound, furthermore ambiguity in word senses. In Natural Processing Language area this usually represents a failure. However, in Computational Humor, it can be seen as an advantage that can be taken as advantage in generation and identification of this kind of texts. In this article the most used linguistic phenomena are described. Additionally an algorithm for homophones phrases generation based on a phrase is described, using a phonetic approach and a non-humorous lexicon, neither trained. This algorithm was able to generate 22 of 45 puns taken, mainly, from popular guessing games.

Keywords: pun, computational humor, homophony.

1. Introducción

Los chistes, normalmente, tienen una preparación y un remate. La preparación crea ciertas expectativas y el remate las rompe, conduciendo así a diferentes interpretaciones de la preparación [4]. Cuando se trata de chistes que involucran juegos de palabras sucede lo mismo. Los chistes de juegos de palabras, o chistes que involucran juegos verbales, son una clase de chistes que dependen de palabras que son similares en sonido, pero se usan con diferentes significados. La diferencia entre los significados crea un conflicto que rompe la expectativa y es humorística. El juego de palabras puede ser creado entre: dos palabras con la misma pronunciación y ortografía, dos palabras con diferente ortografía pero misma pronunciación y, con dos palabras con diferente ortografía y pronunciación similar [5].

Como ejemplo tenemos el chiste “le pedí un café y me dijo ‘sólo queda té’, fue hermoso”. La primera frase “le pedí un café” establece cierta expectativa; la segunda frase “me dijo ‘sólo queda té’” es una respuesta aceptable y encaja con la primera; sin embargo, la última frase “fue hermoso” fuerza a interpretar la segunda de manera diferente, se sitúa ahora en un contexto romántico. La segunda frase, entonces, debe leerse como “sólo quédate” que es homófona a la original. Se ha logrado pues romper la expectativa y se produce así un efecto cómico o divertido.

2. Estado del arte

Investigadores de los tiempos antiguos de Aristóteles y Platón hasta nuestros días se han esforzado por descubrir y definir el origen del humor. Hay casi tantas definiciones del humor como teorías [1]. En el área computacional se han realizado apenas algunos trabajos y pocos han sido en español.

Para el idioma inglés una de las investigaciones más destacadas es el de [5] quienes crearon un programa que reconoce chistes *knock, knock* utilizando técnicas de n-gramas, un generador/reconocedor de juegos de palabras y un reconocedor/generador de remate de chistes. La comicidad de este tipo de chistes se basa en el juego verbal, sobre todo en el sonido de las palabras, principalmente en la paronimia, homonimia y homofonía.

En cuanto a juegos de palabras, se reporta que el programa fue capaz de encontrar el juego de palabras (contenido en la última línea de un tipo específico de chiste *knock, knock*) en 85 de 122 chistes que pudo haber reconocido potencialmente y, en la mayoría de los casos, el juego de palabras encontrado coincidió con el esperado. En cuanto al reconocimiento de chistes, el programa reconoció exitosamente 62 de los no chistes usando n-gramas. De los 130 chistes nuevos (desconocidos para el sistema) el programa no pudo reconocer ocho.

También realizaron algunos experimentos [6] en los que el objetivo era encontrar las palabras que producen el juego de palabras en chistes. Los autores parten de la hipótesis de que todo chiste de juego de palabras contiene un par de palabras *source* y *target*. Indican que *source* es una frase en el chiste que entra en conflicto con la interpretación inicial del perceptor; *target* es una frase que suena similar a *source* y, la similitud entre

ambas, además del conflicto de la interpretación inicial, conduce al descubrimiento de una segunda interpretación del texto. El experimento consistió en buscar la palabra *source* de final a principio del chiste y, una vez que se encontraba, se generaron varias palabras *target*. En resumen, la investigación toma ventaja de la estructura del chiste en relación a su familiaridad y valores de frecuencia KF para aplicar los reconocedores de recursos en la manera más eficiente.

JAPE [2] es un software capaz de generar *punning riddles* a partir de un léxico no humorístico. En ese trabajo se desarrolló un modelo formal de estos acertijos el cual se compone de cuatro partes: esquemas, que especifican las relaciones entre las unidades léxicas utilizadas para construir un chiste; el generador SAD (*Small Adequate Description*) que contiene descripciones cortas del mundo o de unidades léxicas construidas; las plantillas, que convierten unidades léxicas y sus descripciones en un formato de pregunta-respuesta y; recursos léxicos, que proveen toda la información léxica requerida para que las otras partes funcionen.

Los *punning riddles* son un subtipo de humor basado en juegos de palabras con un formato de acertijo. Cuentan con ciertos mecanismos de construcción y estructuras regulares. Un ejemplo de *punning riddle* es “What kind of tree is nauseated? – A sick-amore”. En el idioma inglés existe cierta homofonía entre *sick* y *sycamore*. La primera palabra se usa para decir que una persona sufre de náuseas. La segunda se refiere a un árbol. Se puede decir que *sick* se obtiene de aplicar aféresis a *sycamore*. La similitud fonética entre ambas y el rompimiento de la expectativa creada por la pregunta inicial producen la comicidad del acertijo.

3. Fenómenos lingüísticos involucrados

Algunos juegos de palabras están basados en la homofonía entre dos palabras diferentes en significado y/o diferente ortografía, o bien, entre dos frases que cumplan estas condiciones. De esta manera, los juegos de palabras se prestan a diferentes interpretaciones e incluso a construcciones léxicas diferentes dependiendo de cómo suena la frase. En la mayoría de los casos esta homofonía es producida por los fenómenos lingüísticos que se describen a continuación.

Uno de los fenómenos más comunes en los juegos de palabras (o juegos de frases) es el calambur, el cual consiste en realizar un agrupamiento de sílabas diferente al original para formar una nueva frase. Por ejemplo, la oración “yo lo quito” puede reestructurarse para formar la oración “yo loquito” que adquiere un significado totalmente diferente. La comicidad de los juegos de palabras recae en exponer la frase que inicialmente se encontraba oculta debido a la homofonía y al significado de dicha frase inicial.

Otros de los fenómenos más frecuentes son la sinalefa, la sinéresis y la contracción. Éstos consisten en pronunciar como uno solo dos sonidos contiguos dentro de una palabra (sinéresis) o, del final de una y el principio de otra (sinalefa). La contracción se presenta cuando dos sonidos contiguos iguales se pronuncian como uno solo. Por ejemplo /t//e//ch//o/ en lugar de “te echo” que, en el plano acústico, son difíciles de

desambiguar. Incluso algunos sistemas de reconocimiento de voz tienen problemas para hacerlo.

En los juegos de palabras también deben considerarse los metaplasmos como aféresis y apócope. Y es que no siempre la frase latente coincide completamente con la frase original, ya sea en su forma escrita o a nivel fonético. La aféresis consiste en la desaparición de uno o más fonemas al principio de una palabra mientras que la apócope se realiza al final de una palabra. Como ejemplo tenemos un chiste en el que un panqué de chocolate hincado frente a una fresa le dice “mi corazón chocolate por ti”. El juego de palabras está en usar “chocolate” por “late” siendo la primera el sabor del panqué. Este juego de palabras es lo que hace divertida a la imagen.

Por tanto, para comprender los juegos de palabras hay que realizar una construcción léxica diferente. La nueva frase debe ser homófona a la inicial o muy similar. También puede ser que la representación fonética de esta nueva frase coincida solo en parte a la de la original. Esta última condición correspondería al apócope y aféresis. En el ejemplo de la oración “El lado oscuro” generaríamos frases como “Helado oscuro” o “Helados curo”.

4. Algoritmo para generar frases homófonas

En este trabajo se propone el siguiente algoritmo para generar frases homófonas a partir de una frase de entrada utilizando un lexicón no humorístico. Este proceso consta de tres fases: generación de la cadena fonética, búsqueda fonética a partir de la cadena fonética utilizando el lexicón y, construcción de nuevas frases.

Se debe considerar que las frases resultantes no son validadas a nivel sintáctico después de generarlas y que, entre mayor número de palabras posea el lexicón más palabras serán encontradas, mismas que producirán un número mayor de frases. En la figura 1 se muestra el proceso general para generar frases homófonas a partir de una oración de entrada.

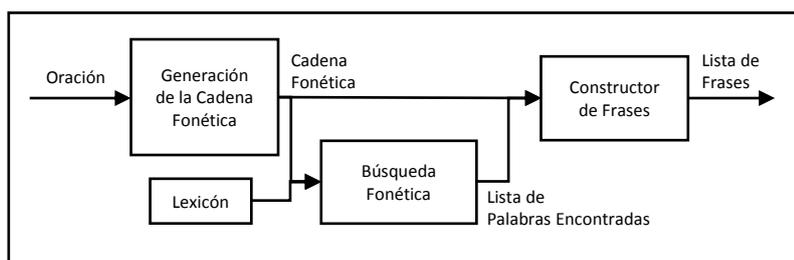


Fig. 1. Proceso general de generación de frases homófonas.

4.1. Generación de la cadena fonética

En esta fase se genera una cadena fonética que representa el sonido de una frase mediante símbolos de un alfabeto fonético. En este caso la oración es una lista ordenada

de palabras. A su vez, una palabra posee los siguientes atributos: forma de palabra, forma fonética, etiqueta gramatical, forma semántica.

Las letras de una oración son símbolos gráficos que corresponden a sonidos del plano acústico. Para transcribir fonéticamente una palabra se aplica una serie de reglas fonológicas que las cambian una o más letras por un símbolo de un alfabeto fonético, éste puede ser tan específico como se desee. En este caso se utiliza una modificación propia de las reglas fonológicas de SAMPA [7]. El motivo de esta modificación es generalizar algunos sonidos (fonemas y alófonos) para producir ambigüedad fonética, y a su vez provocar homofonía entre una palabra y otra o, entre una palabra y una frase o, viceversa.

Una regla fonológica está compuesta de tres elementos: letra(s), reemplazo y contexto. El reemplazo es un símbolo del alfabeto fonético que corresponde al sonido de una o más letras (primer elemento). El contexto es la condición que debe cumplir el primer elemento para ser reemplazado por el segundo.

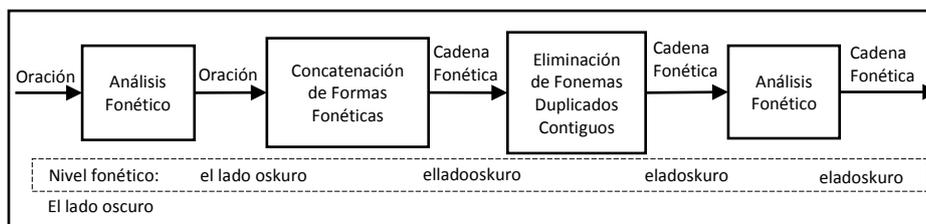


Fig. 2. Generación de la cadena fonética.

Tabla 1. Reglas fonológicas de SAMPA modificadas.

Categorías		
W=ei (vocales débiles)		
F=au (vocales fuertes)		
B=pb (bilabiales)		
Reglas fonológicas ¹		
rr/R/_	q/k/_	z/s/_
r/R/^_	c/s/_W	h/_
ch/X/_	c/k/_	v/b/_
x/X/_	g/j/_W	y/i/_
qu/k/_W	gu_g/_W	ñ/J/_
qu/ku/_F	gü/hu/_	n/m/_B
qü/ku/_	ll/i/_	n/m/_f

En la figura 2 se muestran los sub-procesos que conforman a esta fase y se describen enseguida. Esta sucesión de pasos ayuda a simular fenómenos fonéticos como la contracción, sinéresis y calambur puesto que con la cadena fonética generada no se

¹ El símbolo “^” indica que el primer elemento (la letra r) debe encontrarse al inicio de una palabra.

distinguen los límites de las palabras o *tokens*. Por lo tanto, a partir de ésta se pueden realizar construcciones léxicas diferentes a la oración original.

- Como primer paso se realiza un análisis fonético de la oración de entrada, el cual consiste en enriquecer con su respectiva forma fonética a cada una de las palabras de la oración. Se utilizan las reglas fonológicas de la tabla 1.
- En el segundo paso se genera la cadena fonética concatenando (sin espacios en blanco) cada una de las formas fonéticas de las palabras, respetando su orden de aparición.
- Como tercer paso se eliminan los fonemas duplicados contiguos. Esto con el propósito de simular los fenómenos de contracción, sinéresis y sinalefa. Además de que permite que los sonidos (o fonemas) que se encontraban separados puedan re-analizarse en el paso siguiente.
- Como último paso se realiza un análisis fonético sobre la cadena fonética ya generada. De esta manera, se recodifican los sonidos (o fonemas) que antes del segundo o tercer paso se encontraban separados.

En la frase de ejemplo “el lado oscuro” se generara la cadena fonética “eladoskuro”.

4.2. Búsqueda fonética

Algoritmo 1. Búsqueda fonética.

```
Entrada: cadena_fonética: es una cadena de caracteres previamente generada  
Salida: palabras_encontradas: lista de palabras encontradas  
Descripción: Esta función guarda en resultados las palabras del lexicon cuya forma fonética se encuentre contenida en cadena_fonética. A cada elemento de resultados se le llama palabra encontrada; se compone de una palabra y la posición de inicio y fin de su cadena fonética respecto a cadena_fonética.  
INICIO  
  MIENTRAS (lexicón tenga palabras) HACER  
    palabra = actual palabra del lexicon  
    forma_fonética = forma fonética de palabra  
    SI (cadena_fonética contiene a forma_fonética) ENTONCES  
      iniciaEn = posición de forma fonética en cadena_fonética  
      long = tamaño de cadena_fonética  
      terminaEn = iniciaEn + long  
      crea palabra_encontrada con palabra, iniciaEn, terminaEn  
      añade palabra_encontrada a palabras_encontradas  
    FIN  
  FIN  
  retorna resultados  
FIN
```

La cadena fonética representa el sonido de una frase por medio de símbolos de un alfabeto fonético. En una oración escrita los elementos que la conforman (*tokens*) se encuentran bien definidos y son fáciles de identificar porque se encuentran separados por espacios. En el plano acústico no siempre sucede así, debido a que las personas tendemos a formar vínculos entre el final de una palabra y el inicio de otra. Los límites entre éstas no suelen estar bien definidos. Esto propicia la ambigüedad fonética y esta, a su vez, provoca que estos sonidos puedan interpretarse de una manera diferente a la deseada, es decir, crear nuevas oraciones.

A continuación se genera una lista de palabras encontradas² a partir de la cadena fonética obtenida de la fase anterior. El objetivo es seleccionar aquellas palabras del lexicon cuya forma fonética se encuentra contenida en la cadena fonética. Por ejemplo, una cadena fonética “deskritos” puede interpretarse como “de escritos” o “descritos”.

4.3. Construcción de nuevas frases

Cuando una persona escucha, sin previa contextualización, los sonidos /eladoskuro/ podría entender frases como “el lado oscuro” o “helados curo” o “helado oscuro”. La idea es generar nuevas frases a partir de las palabras encontradas en la fase anterior y la cadena fonética de la primera fase.

Algoritmo 2. Construcción de nuevas frases.

```
Entrada: palabras_encontradas: lista de palabras encontradas ordenadas
Salida: frases: lista de frases generadas
Descripción: genera una lista de frases a partir de una lista de palabras
encontradas y una cadena fonética (misma usada para generar la lista de
palabras encontradas).
INICIO
  MIENTRAS (palabras_encontradas tenga elementos) HACER
    palabra_encontrada_1 = actual palabra de la lista
    crea lista_palabras
    añade palabra_encontrada a lista_palabras
    cadena_fonética_1 = remoción_fonética(cadena_fonética,
palabra_encontrada_1)
    MIENTRAS (palabras_encontradas tenga elementos) HACER
      palabra_encontrada_2 = remoción_fonética(cadena_fonética_1,
palabra_encontrada_2)
      SI (cadena_fonética_1 == cadena_fonética_2) ENTONCES
        añade palabra_encontrada_2 a lista_palabras
        cadena_fonética_1 = cadena_fonética_2
      FIN
    FIN
  ordena lista_palabras por índice inicial, luego por índice final y por
índice inicial
  crea frase con lista_palabras, cadena_fonética
  añade frase a frases
  FIN
RETORNA frases
FIN
```

El proceso consiste básicamente en generar combinaciones de las palabras encontradas, respetando su orden de aparición, verificando que éstas puedan coexistir fonéticamente. Para lograrlo se utiliza la función *remoción_fonética*. En el ejemplo de la cadena fonética “eladoskuro” las palabras encontradas “helado” y “lado” no podrían coexistir porque sus formas fonéticas se traslapan. Esto no significa que una de las dos no pueda usarse en ningún caso sino que deben construirse oraciones diferentes: con la

² Una palabra encontrada se compone de una palabra y posiciones inicio y fin de su forma fonética respecto a una cadena fonética. Por ejemplo la cadena fonética “eladoskuro” contiene la cadena “elado” que es la forma fonética de “helado”. Sus posiciones de inicio y fin son 0 y 5 respectivamente.

palabra “helado” se construye la frase “helado oscuro”, con la palabra “lado” se construye la frase “el lado oscuro”.

Las frases generadas³ en esta fase no son validadas a nivel sintáctico, por lo tanto podrían carecer de sentido. La lista de palabras encontradas que se introducen al algoritmo deben ser previamente ordenadas por índice inicial, luego por índice final y nuevamente por índice inicial.

Al final se tiene una lista de frases nuevas de las que se calcula su cobertura fonética. Esta métrica ayuda a seleccionar, en un análisis posterior, la mejor frase candidata para analizarla o sustituirla por la original, dependiendo de las necesidades. Por ejemplo, si la longitud de una cadena fonética de una frase es 9 y la cadena fonética original es 10 entonces se tiene una cobertura fonética de 0.9.

A continuación se detalla la función *remoción_fonética* que se utiliza en el algoritmo anterior. Ésta función se asegura de remover la cadena fonética la forma fonética de una palabra a excepción de su primer y último carácter. De esta manera se permite que estos fonemas puedan ser reutilizados por otras palabras y así simular fenómenos como: contracción, sinéresis.

Algoritmo 3. Remoción fonética.

```
Entrada: palabra_encontrada, cadena_fonética
Salida: resultado: es una cadena de caracteres
Descripción: retorna cadena_fonética si la misma no contiene a la forma fonética de palabra_encontrada. En caso contrario reemplaza en cadena_fonética la forma fonética de palabra_encontrada por el primero y último carácter de la forma fonética separados por un espacio.
INICIO
  forma_fonética = forma fonética de palabra_encontrada
  reemplazo = forma_fonética
  long = longitud de forma_fonética
  SI (cadena_fonética contiene a forma_fonética) ENTONCES
    SI (long >= 2) ENTONCES
      primero = primer caracter de forma_fonética
      último = último caracter de forma_fonética
      reemplazo = concatenar(primer, " ", último)
    FIN
  resultado = sustitución de forma_fonética por reemplazo en cadena_fonética
  RETORNA resultado
EN CASO CONTRARIO
  RETORNA cadena_fonética
FIN
```

Esta función verifica si una palabra puede añadirse o no a una oración dependiendo de la cadena fonética que se introduce. De ser así se modifica la cadena fonética a manera de preparación para verificar la siguiente palabra. A continuación se muestra un ejemplo de utilización de la función *remoción_fonética*.

³ Una frase se compone de los siguientes atributos: una oración, que es la nueva construcción léxica; una cadena fonética propia, que corresponde a la oración; una cadena fonética de origen, que es la cadena fonética original y; una cobertura fonética, que es una proporción entre la longitud de la cadena fonética propia y la cadena fonética original.

Tabla 2. Ejemplo de uso de la función *remoción_fonética*.

Entradas			Salida
Cadena fonética	Forma fonética	Reemplazo	
<i>eladoskuro</i>	<i>elado</i> (helado)	e_o	e_oscuro
<i>e_oscuro</i>	<i>lado</i> (lado)		e_oscuro
<i>e_oscuro</i>	<i>oscuro</i> (oscuro)	o_o	e_o_o

A diferencia de los casos primero y tercero (renglones), note que en el segundo la salida de la función es igual a la cadena fonética de entrada. Esto significa que la palabra a la que corresponde esta forma fonética no puede coexistir en la oración y se descarta.

5. Pruebas y resultados

Para probar el algoritmo propuesto se utilizaron calambures de adivinanzas populares. Usualmente, son los calambures los que producen la comicidad en las adivinanzas. Por ejemplo, la adivinanza “oro no es, plata no es” contiene un calambur que, a su vez, es la respuesta de la misma. El calambur “plata no es” debe ser reinterpretada como “plátano es” para así responder a la adivinanza.

Tabla 3. Algunas pruebas del algoritmo en calambures de adivinanzas.

Calambur	Resultado esperado	Grupo seleccionado
Espera	Es pera	eh es pera he es pera es pera espera es pera ah es pera ha
Escapa	Es capa	escapa es capa es cap ah es cap ha
Lana baja	La navaja	lana va ja la ana va ja la han ah va ja la han ha va ja la han navaja

Como resultados se esperan obtener varias frases de las que se mide la cobertura fonética. A continuación se muestran algunas de las pruebas realizadas. Las frases que aparecen en la tercera columna fueron elegidas de entre todas las frases candidatas por tener el 100% de cobertura fonética.

En la implementación del algoritmo se utilizó como lexicón el diccionario de Freeling [3] el cual está formado por más de 556'000 entradas o formas de palabra

diferentes. Al analizar las pruebas se observa que cuanto más extenso es el lexicón mayor será el número de palabras encontradas con las que se forman las frases. El tamaño del lexicón también tiene efecto en las oraciones generadas: se producen frases “segmentadas”. Por ejemplo, la frase “el la ado hoz cu uro” es, en efecto, homófona a la cadena fonética “eladoskuro”. Sin embargo no es sintácticamente válida y, por lo tanto, carece de sentido.

Para medir la cobertura se verificó que el resultado esperado se encontrara en el grupo seleccionado mediante el criterio de cobertura fonética. En las pruebas, la frase esperada se encontró en el grupo seleccionado en 22 de 45 calambures. Es decir, se obtuvo un cobertura del 48.88%.

6. Conclusiones

Con el algoritmo presentado en este artículo es posible producir frases homófonas a una frase dada. Esto permite reinterpretar una oración a partir de su representación acústica por medio de un alfabeto fonética. Las frases generadas permiten reanalizar la frase en cuestión para comprender los significados alternos que se manejan sobre todo en textos humorísticos.

Como trabajo futuro se pretende desarrollar un módulo sintáctico como auxiliar en el proceso de construcción de frases. Se contempla la posibilidad de implementar un modelo de n-gramas para calcular la probabilidad de usar una palabra después de otra y así descartar algunas combinaciones indeseables. Otra posible solución es utilizar reglas gramaticales que ayuden a determinar si una palabra pudiera usarse o no después de otra.

En muchos de los casos la incongruencia a nivel semántico es la que provoca que se rompan expectativas en los textos humorísticos. Por lo tanto, no se planea implementar ningún tipo de validación en ese nivel. Este trabajo está enfocado al humor computacional y por tanto se desea mantener esta ambigüedad.

Los calambures se encuentran usualmente en adivinanzas, chistes u otro tipo de textos humorísticos. Por esa razón se considera que es importante poseer la capacidad computacional para resolverlos, sobre todo si se desea alcanzar una comprensión computacional del humor.

En este artículo no se reportan resultados de precisión porque se espera medirla una vez que se haya desarrollado e implementado un módulo sintáctico que complemente el proceso de construcción de frases.

Referencias

1. Latta, R. L.: *The Basic Humor Process: A Cognitive-Shift Theory and the Case against Incongruity*. Berlin-Nueva York: Mouton de Gruyter (1999)
2. Binsted, K.: *Machine humour: An implemented model of puns*. The University of Edinburgh (1996)

3. Carreras, X., Chao, I., Padró, L., Padró, M.: Freeling: An Open-Source Suite of Language Analyzers. Proceedings of the 4th Language Resources and Evaluation Conference (LREC 2004), 4, 239–242 (2004)
4. Ritchie, G.: Developing the Incongruity-Resolution Theory (1976)
5. Taylor, J. M., Mazlack, L. J.: Humorous Wordplay Recognition. In International Conference on Systems, Man and Cybernetics Proceedings (Vol. 4, pp. 3305–3311). The Hague, The Netherlands: IEEE (2004)
6. Taylor, J. M., Mazlack, L. J.: Reverse engineering Humor, 45221 (2008)
7. Wells, J.C.: SAMPA computer readable phonetic alphabet. In Gibbon, D., Moore, R. and Winski, R. (eds.), Handbook of Standards and Resources for Spoken Language Systems. Berlin and New York: Mouton de Gruyter, Part IV, section B (1997)

Metodología para la creación de objetos de aprendizaje adaptables al estilo de aprendizaje

Rodolfo Ibarra-Orozco¹, Aremy Virrueta-Gordillo², Benedicto Ramírez-Santiago¹,
Flavio Castillo-Silva¹

¹ Instituto Tecnológico del Valle de Oaxaca,
México

² Universidad Politécnica de Chiapas,
México

{rodolfo.ibarrao, aremyvirrueta, doctor.flaviocastillo}@gmail.com,
benedicto.ramirez.santiago@hotmail.com

Resumen. En este artículo se presenta una metodología para generar objetos de aprendizaje adaptables al estilo de aprendizaje del estudiante. Se propone una metodología hipertextual adaptativa para clasificar un objeto de aprendizaje (OA), insertar esa clasificación dentro del OA, y presentar el OA al estudiante, dentro de un Sistema de Administración del Conocimiento de acuerdo a sus preferencias de aprendizaje. En una siguiente etapa, nos enfocaremos en clasificar estos elementos de forma automática, utilizando técnicas de inteligencia artificial.

Palabras clave: Objetos de aprendizaje adaptables, estilo de aprendizaje.

A Methodology for the Learning-Style Adaptive Learning Objects Creation

Abstract. In this paper, we present a methodology to generate Adaptive Learning Objects based on the students' learning style. An adaptive hypermedia methodology is applied in order to classify a Learning Object (LO) and, then, the obtained classification is inserted into the LO, so a Learning Management System can present the LOs that correspond with the students learning style preferences. At this current research stage, we are particularly interested in developing a framework to build Adaptive Learning Objects, in which the LO and the students' learning styles are manually classified. In the following stage, we will focus on classifying these elements in an automatic way, by means of machine learning techniques.

Keywords: Adaptive learning objects, learning styles.

1. Introducción

El aprendizaje y el cómo aprenden las personas, ha sido por muchos años motivo de estudio de investigadores educativos de diferentes disciplinas, entre ellas la Psicología y Pedagogía, entre otras, partiendo de la premisa que el ser humano es único e irreplicable llegaron en décadas pasadas a implementarlo en el aula, de ahí consideraron que el proceso de enseñanza – comúnmente dada por el docente – no debiese tasar a todos como iguales y que dicho proceso en los modelos centrados en el estudiante debiese ser adaptable al mismo y no que el estudiante intentase responder a un proceso estandarizado.

Muchos de los fracasos educativos en la modalidad presencial durante muchos años y hasta la fecha han sido en parte debido a que el docente con poco conocimiento en la didáctica considere en un grupo orientar su explicación en el promedio entre los conocimientos de los extremos, para que el de abajo del promedio logre avanzar y por consecuencia en el otro extremo frene el desarrollo de los más avanzados, otra situación que también han hecho de manera arbitraria es considerar que todos los estudiantes aprendiesen igual.

Ahora, si esto sucede en la modalidad presencial, en el e-Learning sucede algo parecido o peor, los procesos formativos en medios virtuales han sido, desde sus orígenes estandarizados, entendiendo esto como el diseño de cursos de manera magistrocentrista (procesos diseñados centrados en el docente o dando preferencia al proceso de enseñanza) siendo la manera de poder ofertar algún proceso formativo para atender necesidades en el ámbito académico, ergológico o cultural.

Tradicionalmente, los cursos e-Learning no consideran al estudiante que estará accediendo a los cursos y que se sentirá desatendido debido a que no todos aprenden igual, por lo que la incorporación de la inteligencia artificial y los estilos de aprendizaje de la Programación Neurolingüística (PNL) al proceso virtual podrá propiciar que los objetos de aprendizaje sean ajustables basados en un análisis en la fase de encuadre del curso o del conjunto de preferencias, por lo que dichos objetos de aprendizaje deberán contener alternativas tanto generales como específicas en los estilos de aprendizaje. Se han producido iniciativas con la finalidad de lograr el objetivo de aprendizaje digital. Por ejemplo, *blended learning* [14], combina cursos dentro del salón de clases con aprendizaje digital; Castillo [7, 9], propone el desarrollo de objetos de aprendizaje para cubrir necesidades específicas del estudiante. Por otro lado, se han establecido estándares para la construcción de objetos de aprendizaje con el fin de facilitar una administración eficiente de los objetos de aprendizaje dentro del sistema de administración del aprendizaje (LMS, por sus siglas en inglés).

El artículo está organizado como sigue. La sección 2 describe la interacción entre el estudiante y los objetos de aprendizaje. La sección 3 describe, de una manera general el estándar LOM. La sección 4 describe los estilos de aprendizaje y los conceptos de hipermedia adaptativa. En la sección 5, se presenta la metodología propuesta para la creación de objetos de aprendizaje adaptativos. Finalmente, se presentan las conclusiones y el trabajo futuro.

2. Interacción entre estudiantes y objetos de aprendizaje

Los objetos de aprendizaje (OA) son la entidad fundamental dentro de los cursos e-learning. El estándar LOM (del inglés Learning Object Metadata), propone la siguiente definición de un OA: Un OA es “cualquier entidad, digital o no digital, que puede sea utilizada, re-utilizada o referenciada con tecnología de soporte al aprendizaje” [4].

Se han propuesto diferentes estándares para crear OA. Estos estándares especifican los OA como una colección de metadatos organizados y permiten a los OA ser accesibles, adaptables, interoperables y reutilizables para cualquier LMS. Algunos estándares para la creación de OA son SCORM (del inglés Sharable Content Object Reference Model), IMS (Learning Design IMS-LD), y LOM. Estos estándares no integran elementos que pueden ser útiles para lograr una interacción entre el objeto de aprendizaje y el estudiante. El estándar IMS-LD considera interacción entre elementos del paquete, pero los LMS disponibles no están listos para integrar tales paquetes dentro de los cursos.

En este trabajo, se utilizan metadatos para describir los elementos de interacción entre el estudiante y el objeto de estilo de aprendizaje con el objetivo de generar objetos de aprendizaje adaptables (OAA).

En ambientes de aprendizaje tradicionales, el contenido de aprendizaje y la interacción con el estudiante, es guiada por un profesor que aplica diferentes metodologías de aprendizaje para lograr el aprendizaje de cada estudiante. En un ambiente de e-learning, los OA deben ser presentados a los estudiantes en una secuencia diseñada cuidadosamente para mantener al estudiante motivado por el contenido y que vean satisfechas sus necesidades de aprendizaje. La presentación del OAA involucra dos etapas: la creación del OAA y la detección del estilo de aprendizaje del estudiante.

Con la finalidad de incluir elementos interactivos en un OA, se especifica una estructura de clasificación. Esta estructura de clasificación toma como base las dimensiones de los sistemas hipertexto adaptativos (SHA), asociando la teoría de estilos de aprendizaje con las dimensiones de los SHA.

Las siguientes secciones describen la clasificación de los OA por estilos de aprendizaje y como los elementos de esta clasificación están integrados dentro de los metadatos.

3. Estándar LOM

El estándar LOM es aceptado generalmente como el estándar para proveer metadatos a recursos multimedia de aprendizaje. La finalidad de utilizar metadatos para describir objetos de aprendizaje es promover el intercambio de material de aprendizaje. LOM utiliza las siguientes categorías para describir recursos.

1. General: Agrupa la información general que describe este recurso como un todo.
2. LifeCycle: Describe la historia y el estado actual de este recurso y aquellos que han afectado este recurso durante su evolución.

3. Meta-MetaData: Describe la información específica sobre los propios metadatos (quien creó estos metadatos, cómo, cuándo, y con qué referencias).
4. Technical: Describe los requerimientos técnicos y las características de este recurso.
5. Educational: Describe las características educativas o pedagógicas clave de este recurso. Esta categoría almacena la información pedagógica esencial a aquellos involucrados en lograr una experiencia de aprendizaje de calidad. La audiencia incluye profesores, administradores autores y estudiantes.
6. Rights: Describe los derechos de propiedad intelectual y las condiciones de uso de este recurso.
7. Relation: Define las relaciones entre este recurso y otros recursos etiquetados. Relaciones múltiples pueden ser soportadas.
8. Annotation: Provee comentarios sobre el sobre el uso educativo de este recurso, quién creó esta anotación y cuándo.
9. Classification: Describe dónde se localiza este recurso dentro de un sistema de clasificación particular. Para definir múltiples clasificaciones, puede haber múltiples instancias de esta categoría.

Estudios sobre el uso real del estándar LOM muestra que los metadatos no son utilizados adecuadamente o no son instanciadas. Esto es debido a la alta complejidad de la especificación. Además, algunos de los valores de los metadatos son subjetivos, así que es difícil asignarles un valor.

4. Sistema para clasificación de objetos de aprendizaje

Para clasificar los OA es necesario definir un sistema de clasificación que integra un estándar de metadatos dentro de los OA. Es importante comprender que un sistema de clasificación divide un dominio de la realidad en series ordenadas de categorías y sub-categorías. En este caso, el dominio es una herramienta de preferencias de aprendizaje y las categorías son los estilos de aprendizaje que definen a la herramienta. También, se hace uso de las dimensiones que definen a los sistemas de hipermedia adaptativos para soportar la identificación de los elementos de los sistemas de clasificación.

4.1. Sistemas de hipermedia adaptativos

Un sistema de hipermedia adaptativa obtiene información acerca de los usuarios y su comportamiento y, de acuerdo a sus necesidades, metas y conocimiento actual, adapta la información para después ser presentada de forma personalizada.

Muchos sistemas toman como base los principios de hipermedia adaptativa. Ejemplos de hipermedia educativa son el sistema tutor ISIS [6], un sistema de aprendizaje adaptable al contexto; el tutor Anatom [5], un tutor inteligente para enseñar anatomía; Shaboo [12], un tutor para enseñar los conceptos básico de programación orientada a objetos; Online SHARP [16], un sistema aplicado a la resolución de problemas matemáticos. Estos sistemas utilizan técnicas de adaptación para presentar la información al usuario.

Las herramientas de detección de estilos de aprendizaje identifican formas en las que una persona prefiere aprender. Estas preferencias se agrupan en estilo y son conocidos como “Estilos de aprendizaje” [11]. Keefe [17], define a los estilos de aprendizaje como las características de los comportamientos cognitivos, afectivos y psicológicos que sirven como indicadores relativamente estables de cómo los estudiantes perciben, interactúan y responden en el ambiente de aprendizaje. Dunn [8], los describe como “... la forma en que un estudiante empieza a concentrarse, procesar y retener información nueva y difícil”. Morales [18], los define como un modelo pedagógico para clasificar los elementos cognitivos asociados al estudiante.

Se han realizado estudios para detectar los estilos de aprendizaje en los alumnos, por ejemplo, la investigación realizada por Fleming [13], la cual es generada por el test de VARK: el cuestionario de Honey-Alonso; y el modelo diseñado por Felder y Silverman [11], el cual fue implementado por Spurlin [12]. Este último modelo parece ser el más apropiado para su uso en sistemas educativos basados en la computadora. La mayoría de los modelos de aprendizaje clasifican a los estudiantes en pocos grupos, mientras el modelo de Felder Silverman (FSLSM, por sus siglas en inglés), los describe en una forma más detallada, distinguiendo cuatro estilos de aprendizaje (dimensiones).

- La primera dimensión distingue entre una forma activa y reflexiva de procesamiento de información. Los estudiantes activos aprenden mejor si trabajan activamente con el material de aprendizaje. En contraste, los estudiantes reflexivos prefieren pensar y reflexionar acerca del material.
- Dimensión de aprendizaje sensitiva-intuitiva. Los estudiantes con preferencia a un estilo de aprendizaje sensitivo gustan aprender hechos y utilizar material de aprendizaje concreto. Estos estudiantes tienden a ser más prácticos que los estudiantes intuitivos y gustan de relacionar el material de aprendizaje con el mundo real. Los estudiantes intuitivos prefieren aprender con material de aprendizaje abstracto. Ellos gustan de descubrir las posibilidades y las relaciones, y tienden a ser más innovadores que los estudiantes sensitivos.
- La tercera, la dimensión visual-verbal, diferencia entre los estudiantes que recuerdan lo que han visto (figuras, diagramas, diagramas de flujo), y los estudiantes que obtienen mayor provecho de una representación textual, sin importar si es escrita o hablada.
- En la cuarta dimensión, los estudiantes son caracterizados de acuerdo a su entendimiento. Los estudiantes secuenciales aprenden en pasos incrementales. En contraste, los estudiantes globales usan un proceso de pensamiento holístico y aprenden a grandes pasos. Ellos tienden a absorber el material de aprendizaje de forma casi aleatoria, sin observar las conexiones, pero, después de aprender suficiente material, repentinamente obtienen el panorama completo.

5. Propuesta de metodología para la creación de objetos de aprendizaje adaptables al estilo de aprendizaje

La metodología propuesta comprende cinco pasos, agrupados en dos procesos:

1. Definición del sistema de clasificación

- o Paso 1. Especificación de las dimensiones de hipermedia adaptativa.
- o Paso 2. Especificación de valores para cada categoría del OA.

2. Clasificación basada en el estilo de aprendizaje e integración con el OA.

- o Paso 3. Seleccionar un estándar de metadatos de OA.
- o Paso 4. Realizar la clasificación del objeto de aprendizaje.
- o Paso 5. Insertar los datos de la clasificación dentro de los metadatos del OA.

En las siguientes dos secciones, se muestra un ejemplo para demostrar como el sistema de clasificación basado en el modelo de estilos de aprendizaje de Felder-Silverman . Primero, el sistema relaciona las dimensiones de hipermedia adaptativa con el modelo de estilos de aprendizaje y, luego, se presenta una propuesta para incluir los datos de clasificación dentro de los metadatos LOM.

5.1. Definición del sistema de clasificación

Paso 1. Especificación de las dimensiones de hipermedia adaptativa. La tabla 1 describe la asociación entre los conceptos de estilos de aprendizaje y las dimensiones de hipermedia adaptativa.

Tabla 1. Dimensiones de hipermedia adaptativa.

Dimensiones de hipermedia adaptativa	Conceptos de estilos de aprendizaje	Ejemplo
¿Dónde pueden ser los sistemas de hipermedia adaptativa útiles?	En un LMS para asegurar que el contenido educativo es presentado de forma adecuada a los estudiantes.	Cualquier LMS como LRN, Moodle, Blackboard.
¿Qué características del usuario son utilizadas como fuente de la adaptación?	El perfil del estudiante detectado.	El modelo de Felder-Silverman define un perfil basado en los estilos AR, SI, VV y SG.
¿Qué puede ser adaptado?	Los objetos de aprendizaje descritos por un estándar y administrados por un LMS.	Objetos de aprendizaje descritos por el estándar LOM.
¿Cuáles son las metas de la adaptación?	Proveer al estudiante con OA asociados a su perfil de aprendizaje.	Desarrollar algoritmos para que el LMS administre la presentación de los OAA.

Paso 2. Especificación de valores para cada categoría del OA. Las categoría del OA corresponden a los estilos de aprendizaje propuestos en el modelo. El rango de valores

para cada categoría está determinado por el nivel al que pertenece el estilo de aprendizaje del estudiante, como se muestra en la Tabla 2 y Tabla 3.

Para este ejemplo se ha definido un sistema de clasificación con cuatro categorías. Estas tienen un valor de un rango definido. El conjunto de valores para cada categoría depende de las características del contenido del OA.

Tabla 2. Categorías de clasificación.

Abreviatura	Descripción
AR	Activo-reflexivo
SI	Sensorial-intuitivo
VV	Visual-verbal
SG	Secuencial-global

Tabla 3. Niveles de pertenencia.

Valor	Nivel de pertenencia
1-3	Activo-reflexivo
5-7	Sensorial-intuitivo
9-11	Visual-verbal

5.2. Clasificación basada en estilos de aprendizaje e integración con el OA

Un OA es considerado como un OAA cuando datos de clasificación son incluidos en la especificación de los metadatos del OA.

Paso 3. Seleccionar un estándar de metadatos. Es necesario seleccionar un estándar de OA para analizar los metadatos en detalle.

En el estándar LOM, existe la categoría *classification*, y esta describe precisamente la pertenencia del OA a un sistema de clasificación particular.

Paso 4. Clasificar el OA.

En este paso se definen los valores a asignar en cada categoría de un OA en particular. En esta primera etapa del proyecto, estos valores son asignados de forma manual a consideración de la persona que desarrolla el OA. Estos OA etiquetados funcionarán como una base de conocimiento para que posteriormente se puedan clasificar los OA de forma automática, esto se puede realizar tomando en cuenta similitudes entre OA y la forma en que son utilizados por el estudiante dentro del LMS.

Paso 5. Insertar los datos de la clasificación dentro de los metadatos del OA.

Este paso consiste en agregar elementos en la categoría seleccionada del estándar.

- 9.2 Taxon Path. Es utilizado para definir la clasificación. Este elemento incluye otros elementos a utilizar: 9.2.1. Source. Indica el nombre de la categoría del sistema de clasificación y 9.2.2. Taxon, que indica el valor de la categoría.
- 9.3 Description. Es utilizada para indicar una descripción del objeto clasificado.

- 9.4 Keywords. Incluye palabras clave para una fácil búsqueda de los OA.

6. Conclusiones

En este artículo se ha presentado una metodología para crear OAA que serán utilizados para la presentación del contenido de los OA de acuerdo a las preferencias de aprendizaje del alumno. La metodología describe cómo definir un sistema de clasificación basado en un modelo de estilo de aprendizaje y explica cómo integrar la clasificación obtenida dentro de un estándar de OA. La clasificación del objeto es sólo una de varias actividades para asegurar que el estudiante tiene acceso a materiales que se ajustan a sus preferencias de aprendizaje en cursos en línea. En esta primera etapa del proyecto se realiza la clasificación de forma manual. Esto es necesario para crear una base de conocimiento que permitirá desarrollar un algoritmo inteligente para identificar el estilo de aprendizaje de forma automática.

Referencias

1. J. Akeroyd: Information management and e-learning: Some perspectives. *Asli proceedings*, volume 57, pages 157–167, Emerald Group Publishing Limited (2005)
2. S. Alexander: E-learning developments and experiences. *Education + Training*, 43(4/5):240–248 (2001)
3. C. Alonso, D. Gallego: CHAEA: Honey-Alonso Learning Styles Questionnaire. Bilbao: Mensajero (1999)
4. C. Arteaga, R. Fabregat: Integración del aprendizaje individual y del colaborativo en un sistema hipermedia adaptativo. *JENUI*, 2(2):107–114 (2002)
5. I.H. Beaumont: User modelling in the interactive anatomy tutoring system ANATOM-TUTOR. *User Modeling and User-Adapted Interaction*, 4(1):21–45 (1994)
6. P. Brusilovsky and L. Pesin. ISIS-Tutor: An adaptive hypertext learning environment. In *Proceedings of JCKBSE*, volume 94, pages 10–13 (1994)
7. L. Castillo, L. Morales, A. González-Ferrer, J. Fernández-Olivares, Ó. García-Pérez: Knowledge engineering and planning for the automated synthesis of customized learning designs. *Current Topics in Artificial Intelligence*, pages 40–49 (2007)
8. R. Dunn: Understanding the dunn and dunn learning styles model and the need for individual diagnosis and prescription. *Reading, Writing, and Learning Disabilities*, 6(3):223–247 (1990)
9. J. Fdez-Olivares, L. Castillo, O. García-Pérez, F. Palao: Bringing users and planning technology together. *Experiences in SIADEX*. In *Proc ICAPS*, pages 11–20 (2006)
10. R.M. Felder, L.K. Silverman: Learning and teaching styles in engineering education. *Engineering education*, 78(7):674–681 (1988)
11. R.M. Felder, L.K. Silverman, B.A. Solomon: Index of learning styles (ILS). North Carolina State University (1999)
12. R.M. Felder, J. Spurlin: Applications, reliability and validity of the Index of Learning Styles. *International Journal of Engineering Education*, 21(1):103–112 (2005)
13. N.D. Fleming: I'm different; not dumb. Modes of presentation (VARK) in the tertiary classroom. In *Research and Development in Higher Education, Proceedings of the 1995*

- Annual Conference of the Higher Education and Research Development Society of Australasia (HERDSA), HERDSA, volume 18, pages 308–313 (1995)
14. Charles R. Graham: Blended learning systems: Definition, current trends, and future directions. Pages 3–21, Pfeiffer Publishing (2005)
 15. A. Gunasekaran, R.D. McNeil, D. Shaul: E-learning: research and applications. *Industrial and Commercial Training*, 34(2):44–53 (2002)
 16. R.R. Hernández, A.B.G. González, F.J.G. Peñalvo, R.L. Fernández: Sharp online: Sistema hipermedia adaptativo aplicado a la resolución de problemas matemáticos. IX Congreso Internacional Interacción, pages 271–284, June (2008)
 17. J.W. Keefe: Assessing student learning styles: An overview. *Student learning styles and brain behavior*, pages 43–53 (1982)
 18. L. Morales, G. Roig: Connecting a technology faculty development program with student learning. *Campus-Wide Information Systems*, 19(2):67–72 (2002)

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
septiembre de 2016
Printing 500 / Edición 500 ejemplares

